EEE319 Optimisation
Lecture 2 Gradient Descent Optimisation – Optimisation from an Example

Prof. Xinheng Wang

xinheng.wang@xjtlu.edu.cn

Office: EE512

Outline

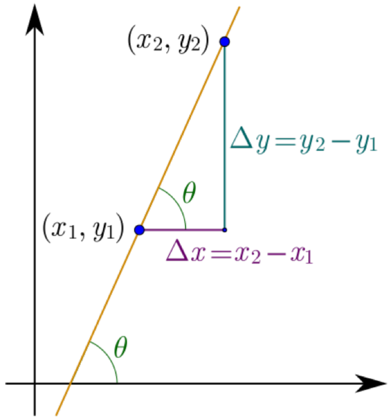
- Basic concepts
- What is gradient descent
- Cost function
- Algorithm implementation

Definition of Gradient Descent

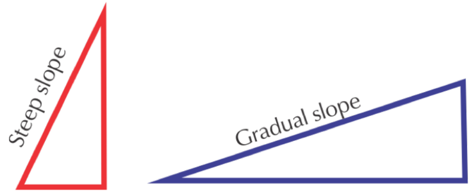
- Gradient by Cambridge Dictionary
  - How steep a slope is (steep vs gradual)
  - In mathematics, the **slope** or **gradient** of a linear function (line) is a number that describes both the *direction* and the *steepness* of the line.
  - Slope is often denoted by letter *m*

$$y = mx + b$$

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$



- Descent by Cambridge Dictionary
  - A movement down



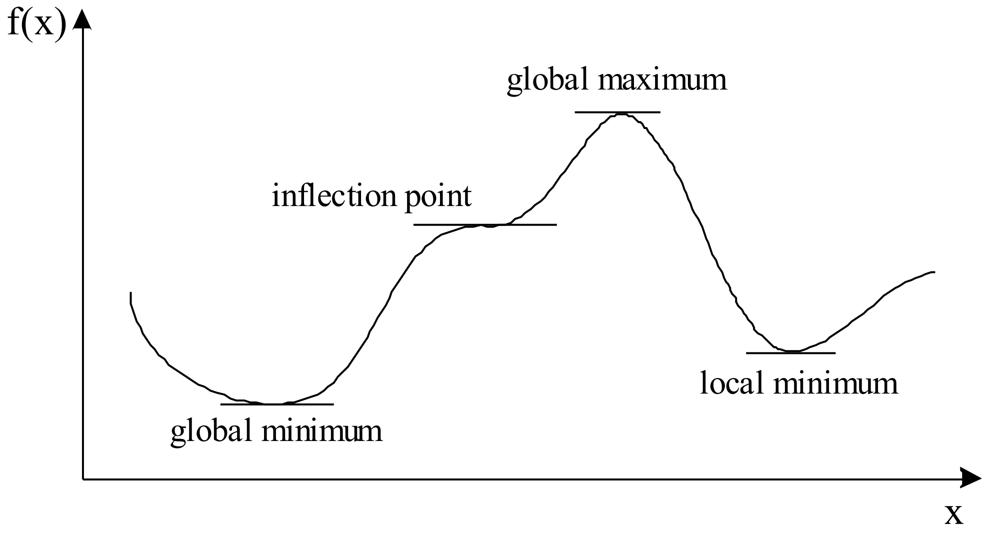
Gradient

- Another way to describe the slope is derivative.
- Given a general function $y = f(x)$, the derivative can be noted as
- $f'(x)$,f',y', df/dx, dy/dx
- Read as "the derivative of *y* with respect to *x*" or less formally, "the derivative of the function."

Maximum or minimum of a function

- To find the maxima and minima of the function, we make
- $\frac{dy}{dx} = 0$

- The least of all the minimum points is called the “global” minimum.
- Every minimum is a “local” minimum
- Not all turning points are minima or maxima

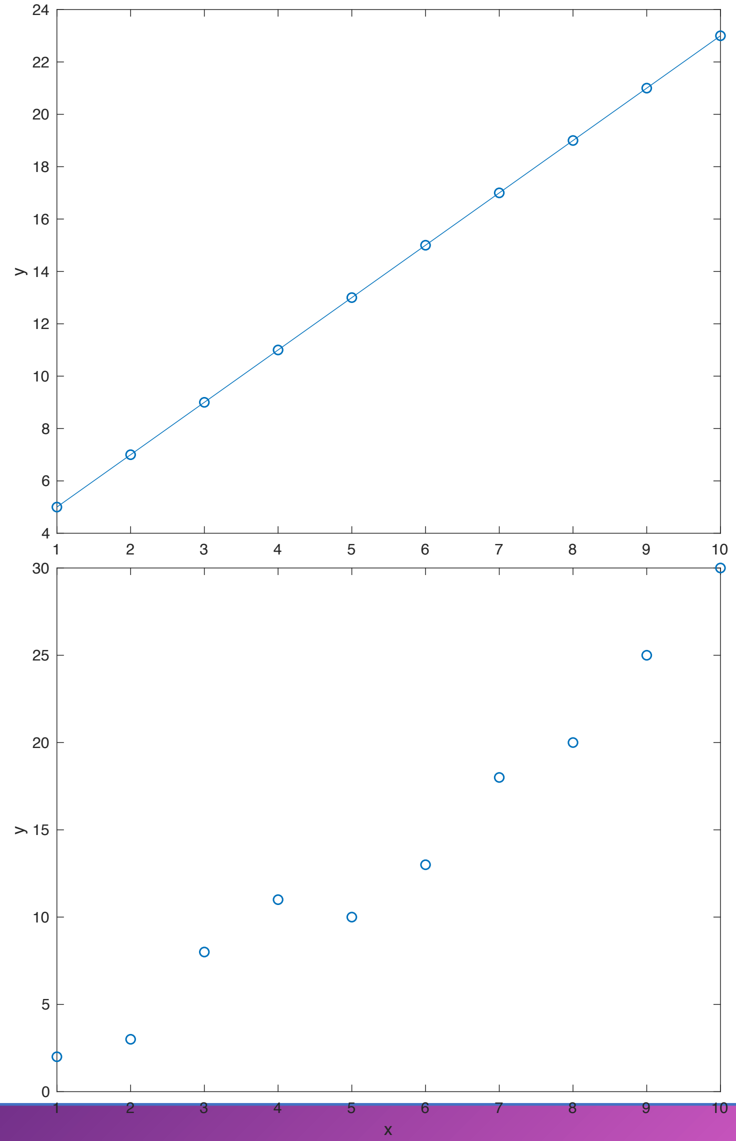


Objective of gradient descent

- To find the minimum of an objective function, by moving downward direction
- But how to do it? Can we just use $\frac{dy}{dx} = 0$ to find the minimum?

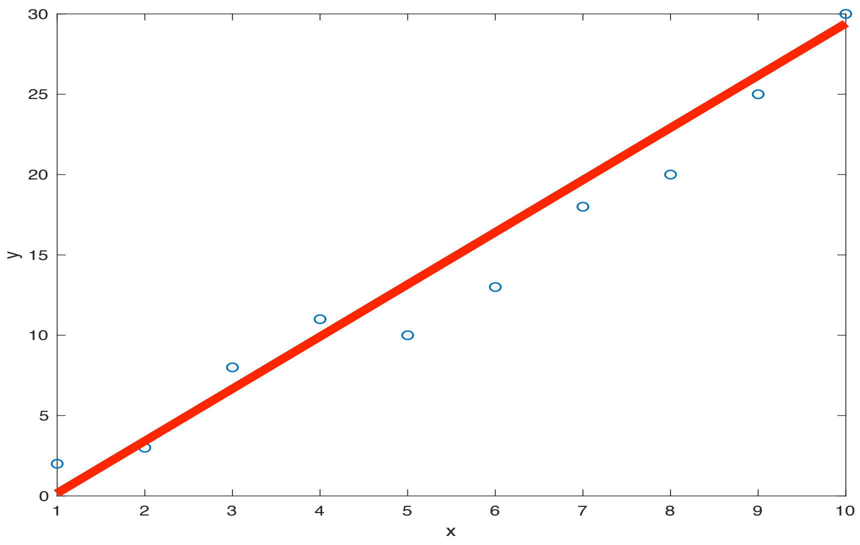
Gradient descent from an example

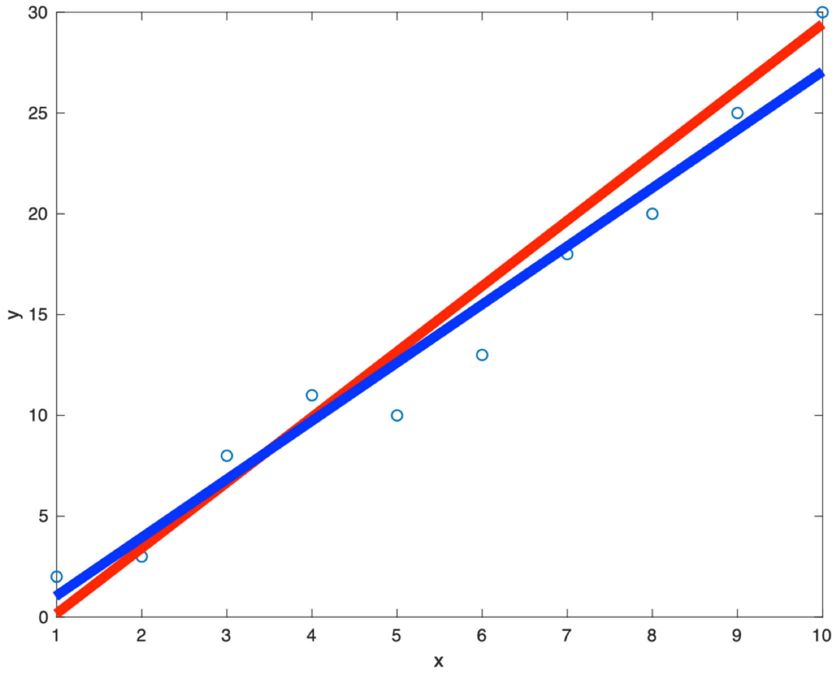
- Taken the linear function $y = mx + b$
- If $m = 2$, $b = 3$ are given, *x*=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, we can quickly draw a line , shown in the figure
- However, if the values of *x*, *y* pairs are given as *x*=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and *y*= [2, 3, 8, 6, 10, 13, 18, 20, 25, 30], how to draw a line to fit these points?



Gradient descent from an example

- Your line could be red one or blue one?



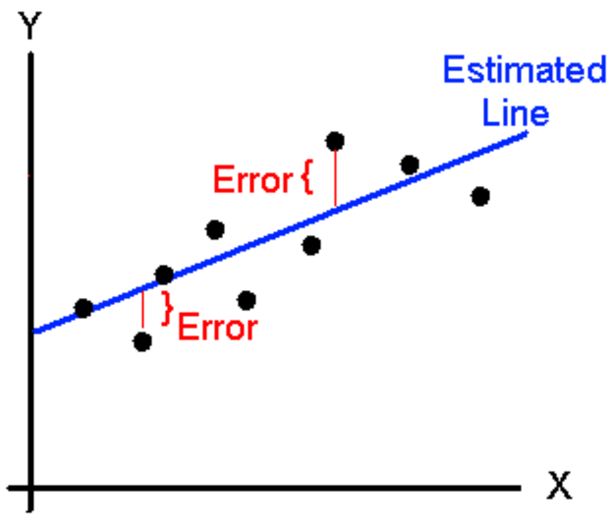


Gradient descent from an example

- Starting from an estimated line, blue line in the figure, for any given value x_i, we will have an estimated value $y_i = mx_i + b$. If the real value is denoted as $\hat{y}_i$, there is an error between these two value.
- How to measure the errors of all points?

$$Error = \frac{1}{N}\sum_{i}^{N}(\hat{y}_i - y_i)^2$$

- where N is the total number of points
- This error is related to the cost function/loss function



Gradient descent from an example

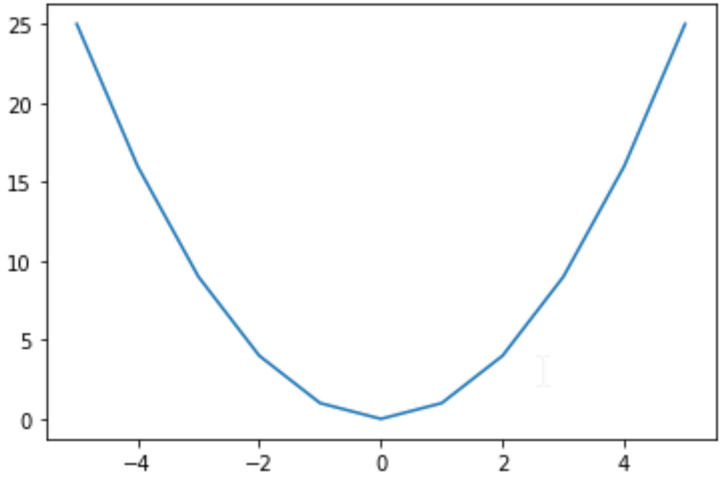
- Cost function/loss function
  - The **Loss function** computes the error for a single training example while the **Cost function** is the average of the loss functions for all the training examples.
- Why using squared errors?
  - Always positive, to avoid cancelling with each other with positive and negative value
  - Emphasising outliers, while neglecting smaller errors
- Aim: Minimise the cost function

Gradient descent from an example

- Look at the equation

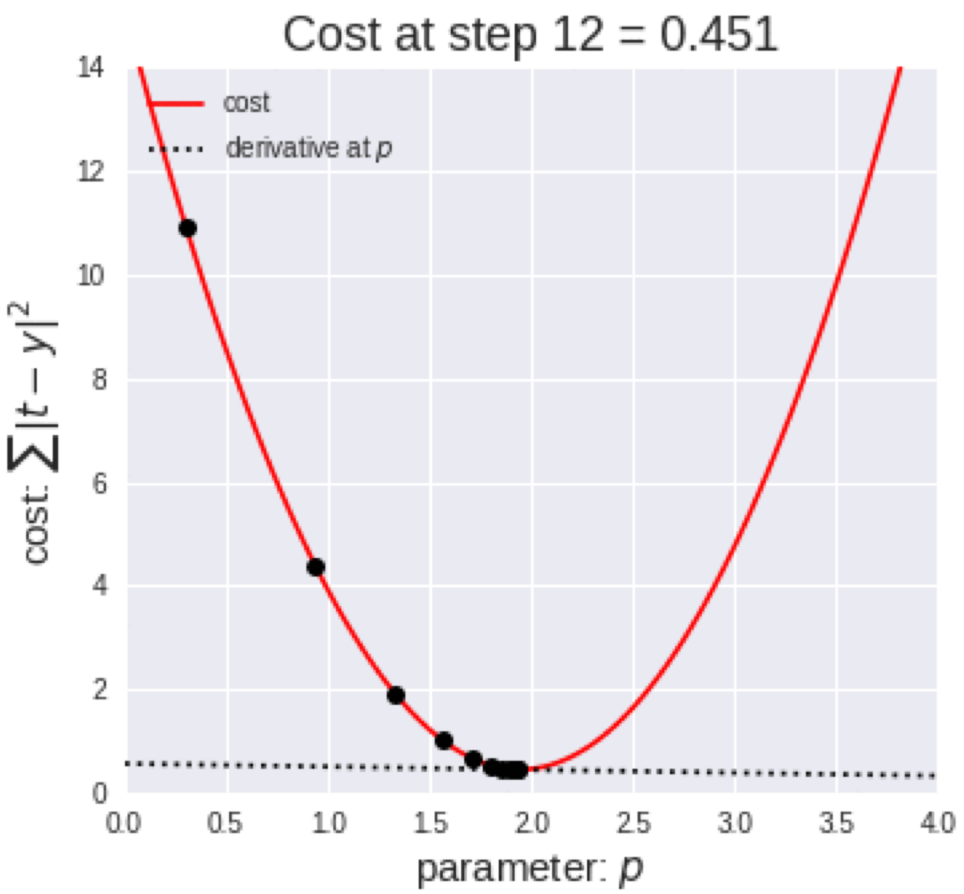
$$Error = \frac{1}{N}\sum_{i}^{N}(\hat{y}_i - y_i)^2$$

- Let us simply the red as X^2, the shape is
- We can see that the minimum is at the bottom.
- If you cannot see this, how can you find the minimum? *m* and *b* also need to be determined.



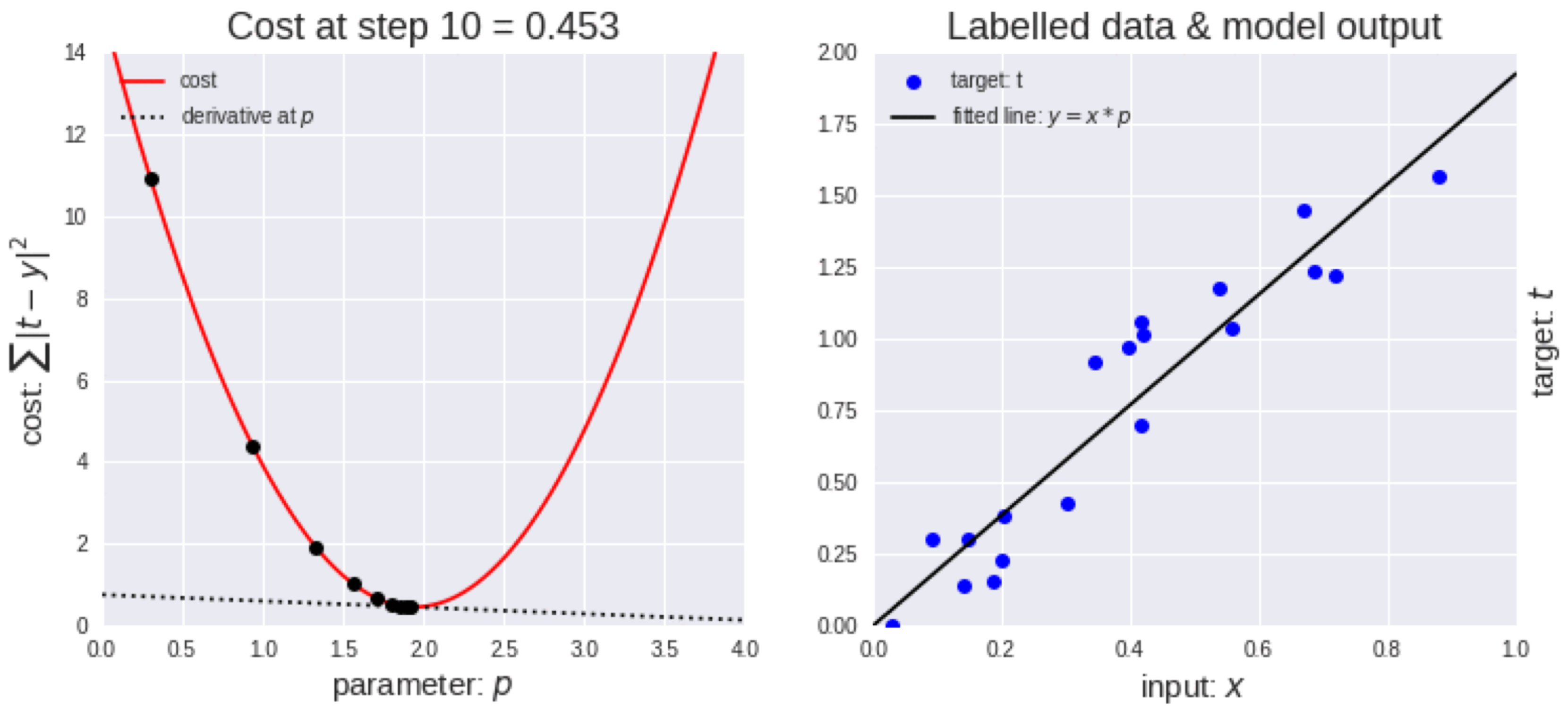
Gradient descent from an example

- An illustration



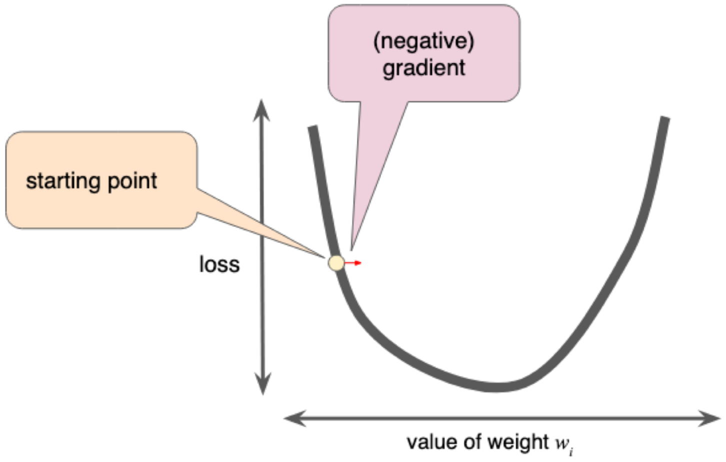
Gradient descent from an example

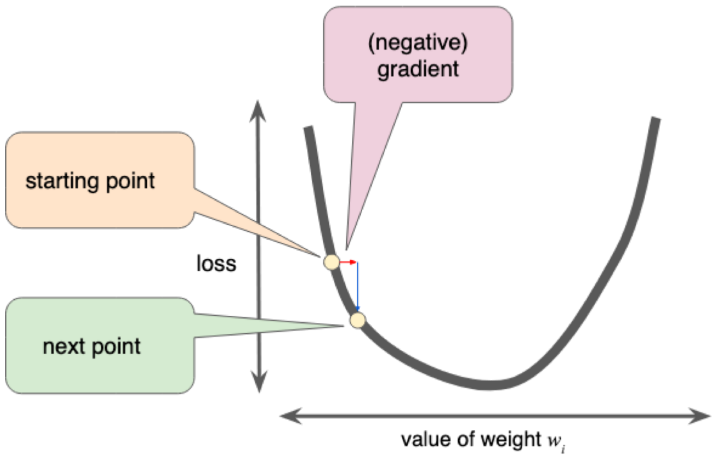
- An illustration of obtaining the minimum and the best fit line



Gradient descent from an example

- Gradient descent finds the minimum (local) of a function by moving along the direction of steep descent (downwards).
- To get to the local minima we can't just go directly to the point. We need to descend in smaller steps and check for minima and take another step to the direction of descent until we get our desired local minimum.





Gradient descent to find the minimum

- Gradient
  - Error function $Error = \frac{1}{N}\sum_i^N(\hat{y}_i - y_i)^2 = \frac{1}{N}\sum_i^N(\hat{y}_i - (mx_i + b))^2$
  - Take the partial derivative of this function with respect to *m* and *b*, respectively

$$D_m = d(\frac{1}{N}\sum_i^N(\hat{y}_i - (mx_i + b))^2)/dm$$

$$D_b = d(\frac{1}{N}\sum_i^N(\hat{y}_i - (mx_i + b))^2)/db$$

Gradient descent to find the minimum

- Gradient
  - Recall $\frac{d}{dx}\sum af_i = \frac{a\sum df_i}{dx}$
  - The above two equations will become

$$D_m = d(\frac{1}{N}\sum_i^N(\hat{y}_i - (mx_i+b))^2)/dm = \frac{1}{N}\sum_i^N d((\hat{y}_i - (mx_i+b))^2)/dm$$

$$D_b = d(\frac{1}{N}\sum_i^N(\hat{y}_i - (mx_i+b))^2)/db = \frac{1}{N}\sum_i^N d((\hat{y}_i - (mx_i+b))^2)/db$$

  - Let $\mathrm{Y} = \hat{y}_i - (mx_i+b)$

$$\frac{1}{N}\sum_i^N d((\hat{y}_i - (mx_i+b))^2)/dm =$$

$$\frac{1}{N}\sum_i^N \frac{d(\hat{y}_i-(mx_i+b))^2}{dY} * \frac{dY}{dm} = \frac{2}{N}\sum_i^N(\hat{y}_i-(mx_i+b))(-x_i) = \frac{-2}{N}\sum_i^N x_i(\hat{y}_i - (mx_i+b))$$

Gradient descent to find the minimum

- Let $\mathrm{Y} = \hat{y}_i - (mx_i + b)$

$$D_m = \frac{1}{N}\sum_{i}^{N} d((\hat{y}_i - (mx_i + b))^2)/dm \; = \frac{-2}{N}\sum_{i}^{N} x_i(\hat{y}_i - (mx_i + b))$$

$$D_b = \frac{1}{N}\sum_{i}^{N} d((\hat{y}_i - (mx_i + b))^2)/db \; = \frac{2}{N}\sum_{i}^{N} (\hat{y}_i - (mx_i + b)) * (-1)$$

Gradient descent

- Now let us use the gradient
- Gradient descent minimises a function by iteratively moving a little bit in the direction of negative gradient, where negative gradient is a vector pointing at the greatest decrease of a function. It is represented mathematically as:

$$x_{i+1} = x_i - a\nabla f(x_i)$$

where a is called learning rate, which is normally a constant, and ∇f is the derivative.

Gradient descent

- Therefore, m and b can be obtained

$$m_{i+1} = m_i - aD_m{}^i$$
$$b_{i+1} = b_i - aD_b{}^i$$

Implementation of Gradient descent

- Code on Matlab
  - Mesh_grid.m
    - Sleeba Paul, Gradient Descent Algorithm with Linear Regression on single variable, https://uk.mathworks.com/matlabcentral/fileexchange/56297-gradient-descent-algorithm-with-linear-regression-on-single-variable

Summary

- Optimisation from an example
- No constraints
- No confirmation of size of errors yet

THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學