Deep Network Development

Lecture #2

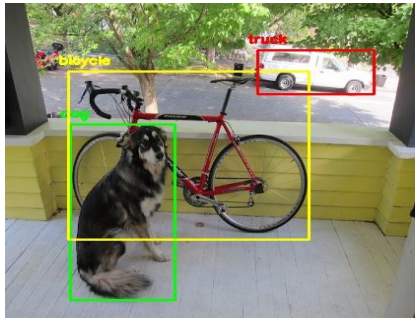
Viktor Varga
Department of Artificial Intelligence, ELTE IK

Image sources: boardgamegeek.com

Last week - Motivation for machine learning

Writing an algorithmic solution might not be ideal when…

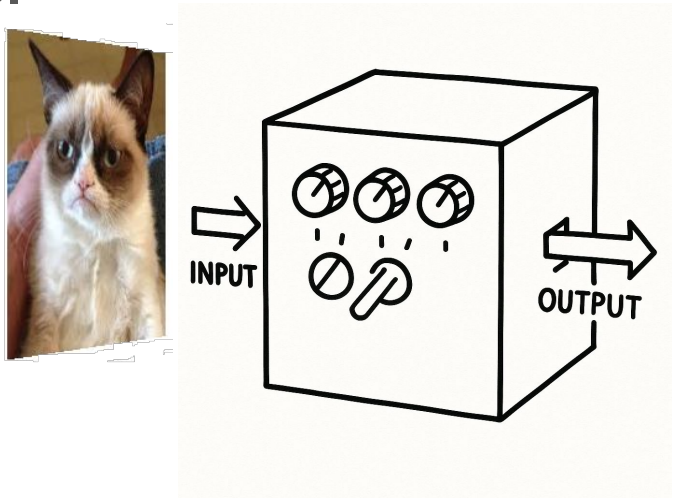
1. We need to solve **many** slightly different **varieties** of the same task.
2. An **algorithmic solution is unknown** or hard to compute.
3. The task can **only be formulated by** showing **examples**



Last week - The essence of machine learning

We **learn to solve the problem** without knowing the specific algorithm for the solution.

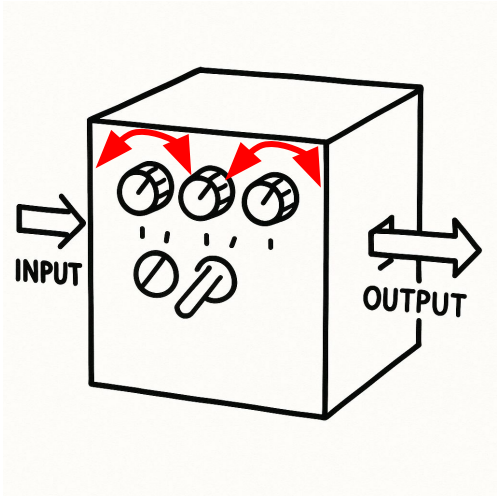
- We define a machine learning **model with parameters**.
- We try to **find such parameters** that will make our model **solve the task** as **well** as possible.
- We **define the task** and its correct solution **by showing examples**.



Last week - The essence of machine learning

We **learn to solve the problem** without knowing the specific algorithm for the solution.

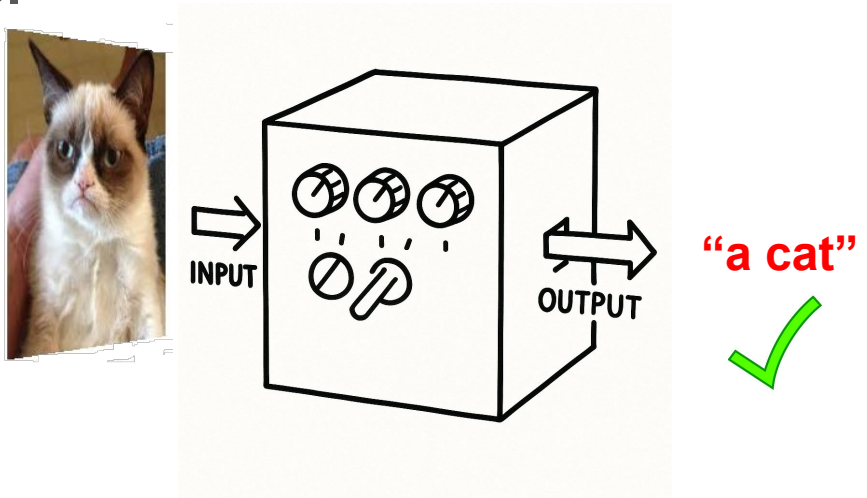
- We define a machine learning **model with parameters**.
- We try to **find such parameters** that will make our model **solve the task** as **well** as possible.
- We **define the task** and its correct solution **by showing examples**.



Last week - The essence of machine learning

We **learn to solve the problem** without knowing the specific algorithm for the solution.

- We define a machine learning **model with parameters**.
- We try to **find such parameters** that will make our model **solve the task** as **well** as possible.
- We **define the task** and its correct solution **by showing examples**.



Last week - Machine learning

Three main groups of machine learning methods:

- **Supervised learning**
- Unsupervised learning
- Reinforcement learning

Last week - Supervised learning

Given: The training sample, a set of (input, label) pairs

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$$

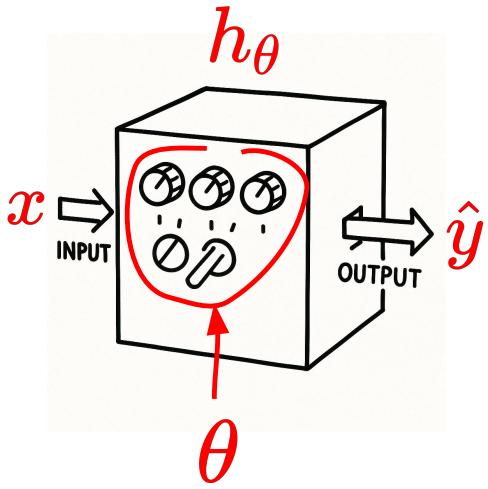
$$x \in X \subset \mathbb{R}^n,\ y \in Y \subset \mathbb{R}^k$$

Task: The estimation of the label (the expected output) from the input

I.e., we search for a (hypothesis-)function h_θ, for which:

$$h_\theta(x) = \hat{y} \approx y$$

Last week - Supervised learning



Given: The training sample, a set of (input, label) pairs

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$$

$$x \in X \subset \mathbb{R}^n, \; y \in Y \subset \mathbb{R}^k$$

Task: The estimation of the label (the expected output) from the input

I.e., we search for a (hypothesis-)function h_θ, for which:

The predicted label

$$h_\theta(x) = \hat{y} \approx y$$

The true label

Goal: Find θ that makes $\hat{y}$ be as similar to y as possible!

The two main types of tasks in supervised learning

Regression: Continuous labels (The label set is infinite)

$|Y| = \infty$ **Example:** Number of cars or the age of a person

Classification: Discrete labels (The label set is finite)

$|Y| < \infty$ **Example:** Categorization of examples

- What is the profession of the person in the image?

The two main types of tasks in supervised learning

Regression: Continuous labels (The label set is infinite)

$|Y| = \infty$ **Example:** Number of cars or the age of a person

Classification: Discrete labels (The label set is finite)

$|Y| < \infty$ **Example:** Categorization of examples

- What is the profession of the person in the image?

Regression

Regression: Continuous labels (The label set is infinite)

$$|Y| = \infty$$

Example: Number of cars or the age of a person

Regression

Regression: Continuous labels (The label set is infinite)

$$|Y| = \infty$$

Example: Number of cars or the age of a person

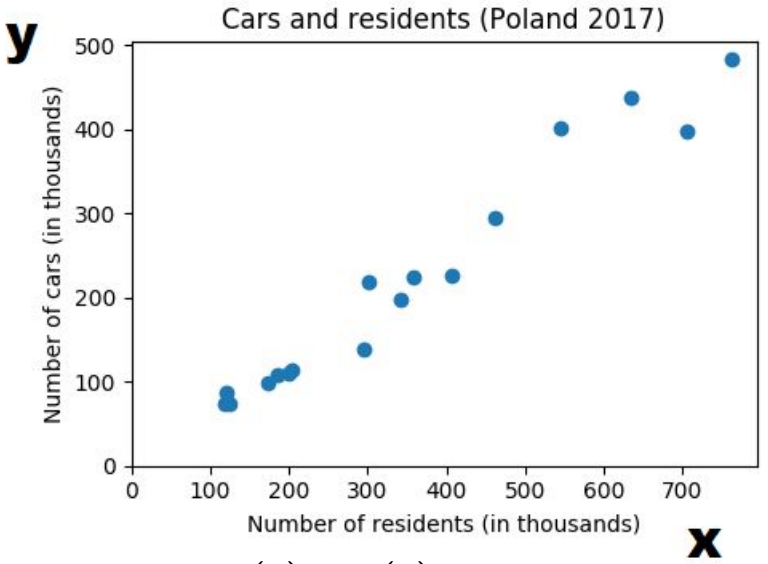
Today: A simple method to solve regression problems.

Regression

Example: Estimate the number of cars in a particular city given the population of that city.

x: The **population** of a particular city

y: The **number of cars** in a particular city



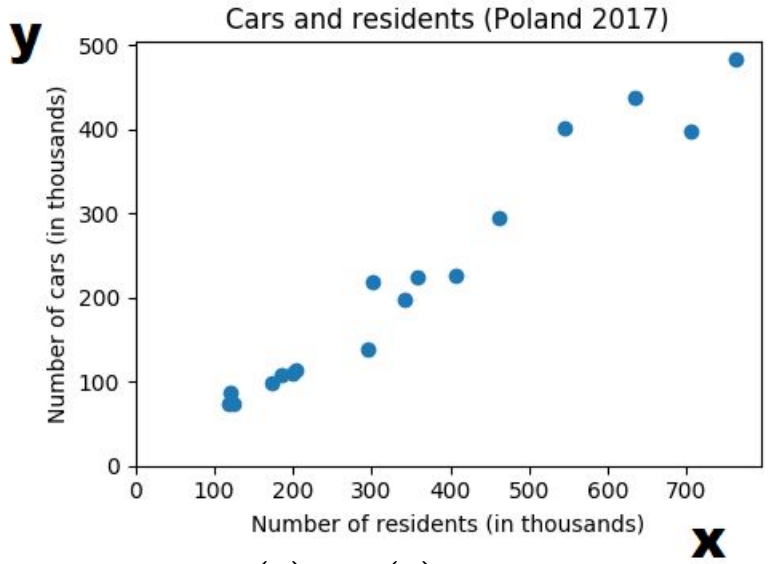
$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Regression

Example: Estimate the number of cars in a particular city given the population of that city.

x: The **population** of a particular city

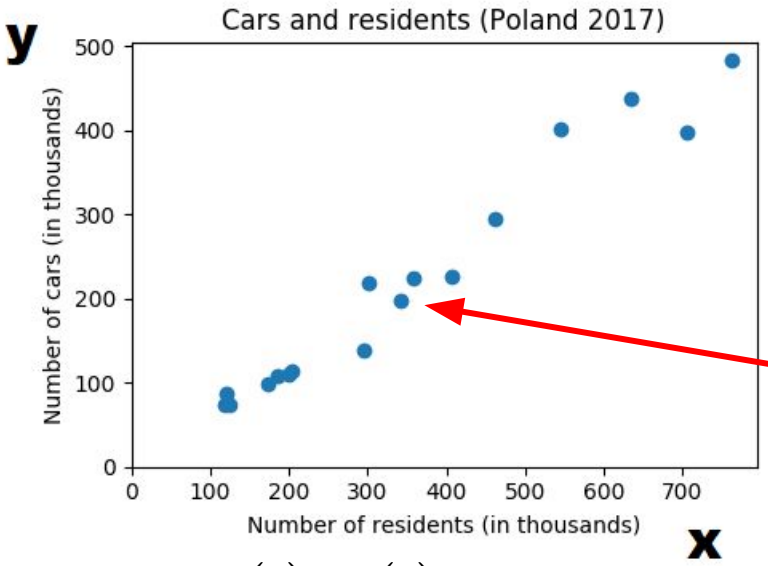
y: The **number of cars** in a particular city



A single input variable and **a label:**
Our labeled examples can be interpreted as points located in a two-dimensional vector space (a plane).

$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

Regression



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

	the input variable (feature) x (population, ×1000)	the label y (n. of cars, ×1000)
Warsaw (j = 1)	1760	910
Krakow (j = 2)	770	465
Lublin (j = 3)	340	198
...	...	...

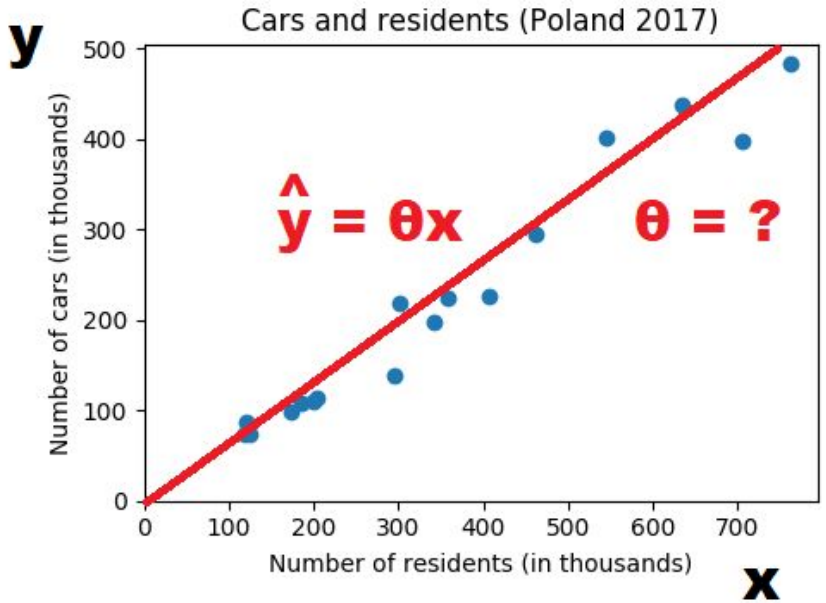
Linear regression

Hypothesis function for linear regression (not yet complete!)

A very simple (linear)
hypothesis function:

$$y \approx \hat{y} = h_\theta(x) = \theta x$$

θ is the parameter of the hypothesis-function: in this case,
this is going to be the **slope of the line**.



Linear regression

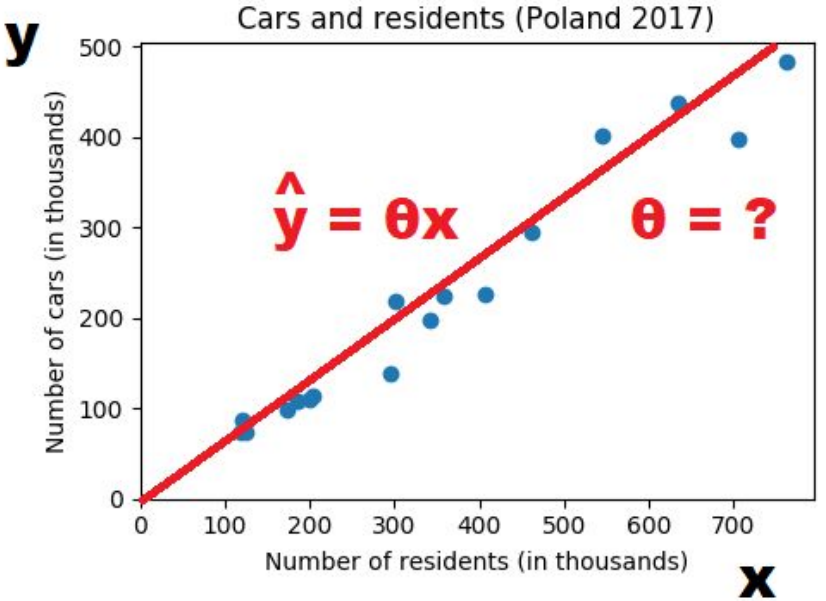
Hypothesis function for linear regression (not yet complete!)

A very simple (linear) hypothesis function:

$$y \approx \hat{y} = h_\theta(x) = \theta x$$

θ is the parameter of the hypothesis-function: in this case, this is going to be the **slope of the line**.

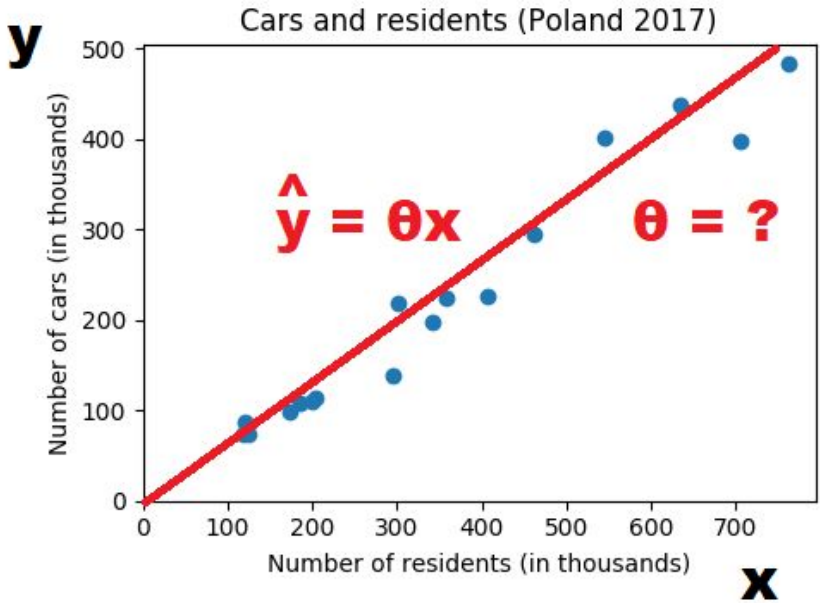
We are looking for a parameter θ such that h(x) closely approximates the true y labels.
For example, the hypothesis function h(x) = 0.65*x fits this particular sample well, so a good parameter is θ = 0.65.



Linear regression

How do we determine how good the estimate is?

$$h_\theta(x) = \hat{y} \overset{?}{\approx} y$$



Supervised learning - Loss function

How do we determine how good the estimate is?

$$h_\theta(x) = \hat{y} \overset{?}{\approx} y$$

With the help of the loss function J: $J : \theta \to \mathbb{R}_{\geq 0}$

The loss function indicates **how much the actual label differs from our estimate** for given parameter values.

Supervised learning - Loss function

How do we determine how good the estimate is?

$$h_\theta(x) = \hat{y} \overset{?}{\approx} y$$

The loss must be a single number (scalar). If the y label consists of multiple variables, then instead of the absolute value, we will need, for example, a norm...

With the help of the loss function J:

$$J : \theta \to \mathbb{R}_{\geq 0}$$

The loss function indicates **how much the actual label differs from our estimate** for given parameter values.

The greater the error in our estimate
with a given parameter **θ**, the greater the loss is in **θ**.
In case of most loss functions: a loss value of 0 indicates a perfect estimate.

Linear regression - The least squares method

When using **the least squares method**, our loss function is:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\overbrace{h_\theta(x^{(j)})}^{\hat{y}^{(j)}} - y^{(j)})^2$$

We define the loss as the **squared differences** of true labels and estimates.

Linear regression - The least squares method

When using **the least squares method**, our loss function is:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (h_\theta(x^{(j)}) - y^{(j)})^2$$

We define the loss as the **squared differences** of true labels and estimates. We take the **mean of these errors** over the training dataset.

Linear regression - The least squares method

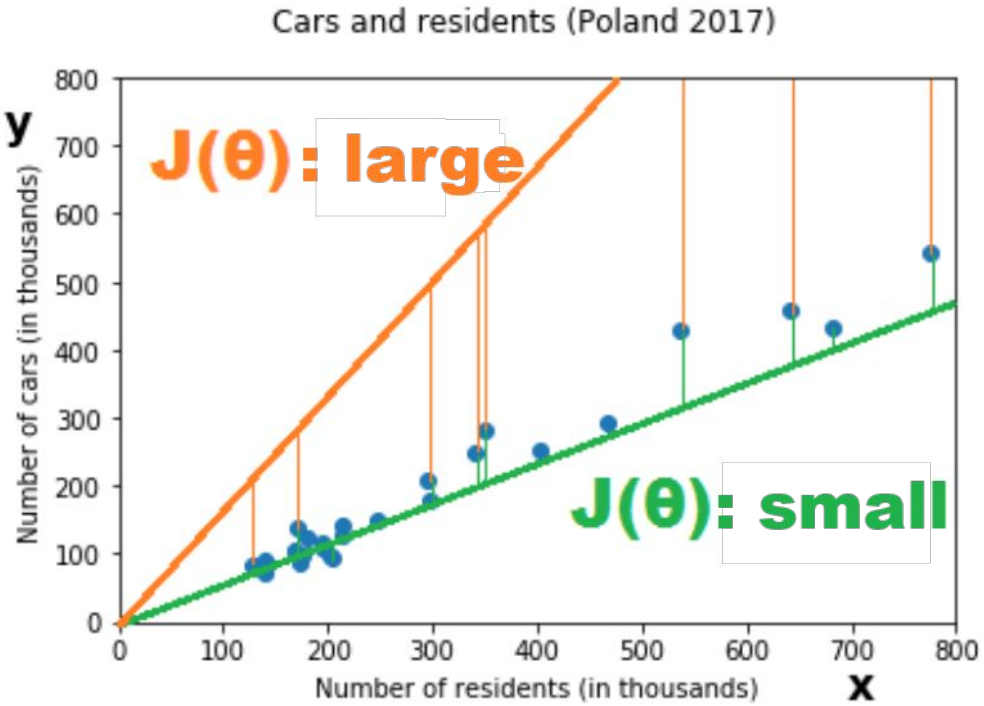
When using **the least squares method**, our loss function is:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (h_\theta(x^{(j)}) - y^{(j)})^2$$

We define the loss as the **squared differences** of true labels and estimates. We take the **mean of these errors** over the training dataset.

→ **Mean Squared Error (MSE) loss**

Linear regression - The least squares method



Linear regression - The least squares method

Our goal: $$\theta^* = \underset{\theta}{\operatorname{argmin}}\, J(\theta)$$

→ We search for the **optimal parameter (θ*)** that minimizes the loss, i.e., the mean squared error of our label prediction from the true label.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{2m} \sum_{j=1}^{m} (h_\theta(x^{(j)}) - y^{(j)})^2 =$$

$$= \underset{\theta}{\operatorname{argmin}} \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

Linear regression - The least squares method

Our goal: $\theta^* = \underset{\theta}{\operatorname{argmin}}\, J(\theta)$

x, **y** are known constants.
They simply come from the dataset.
We are looking for a good θ.

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

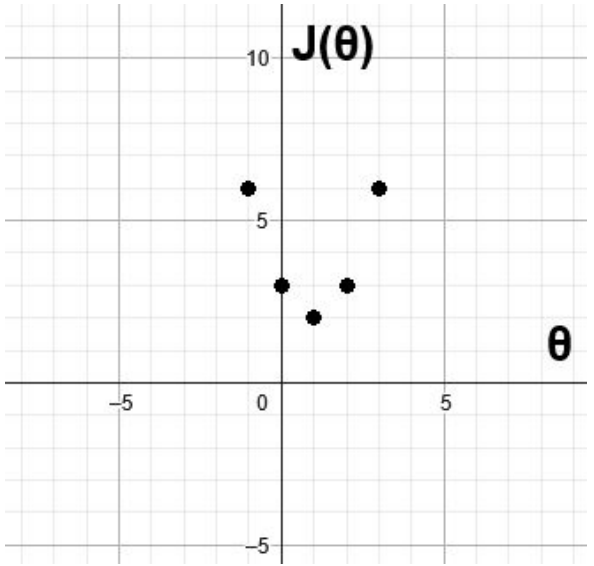
How do we find the optimal parameter θ*?

Linear regression - The least squares method

How do we find the optimal parameter θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ values (parameters)!

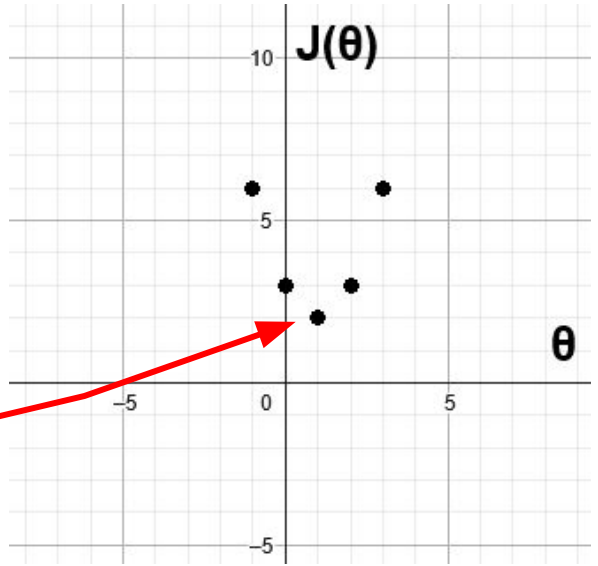


Linear regression - The least squares method

How do we find the optimal parameter θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ values (parameters)!



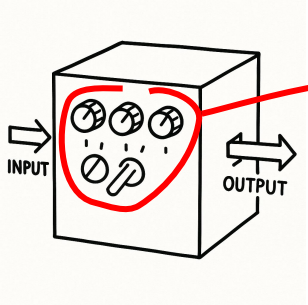
For example, here, from the parameter values θ = {-1, 0, 1, 2, 3},
the loss is the smallest in θ = 1

Linear regression - The least squares method

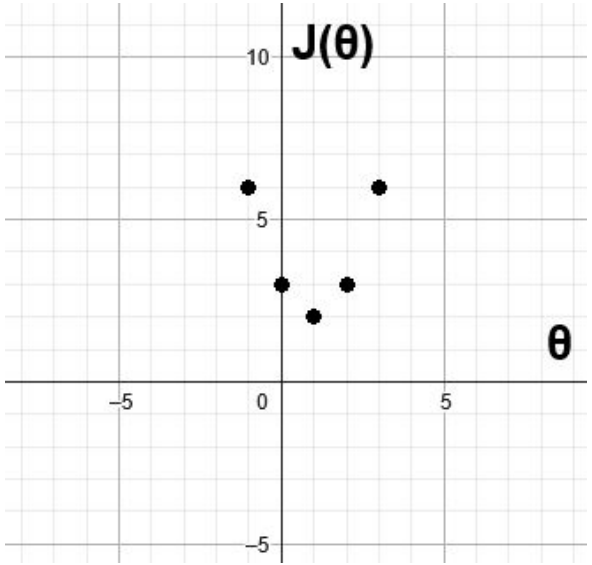
How do we find the optimal parameter θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ values (parameters)!



Later, we will have
more than one parameters…

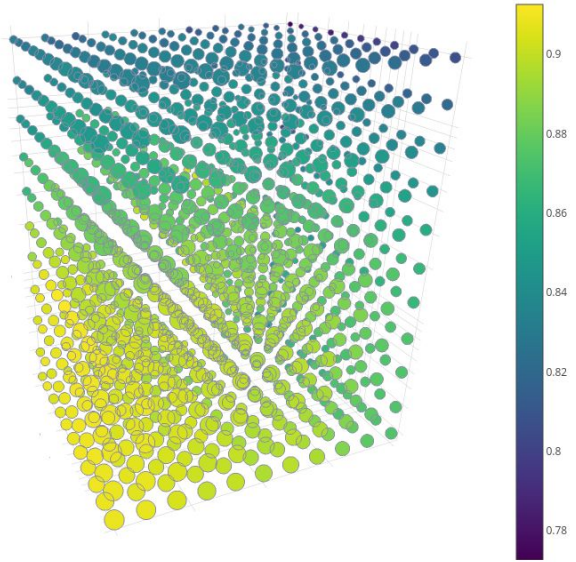


Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ **points** (parameter **combinations**)!

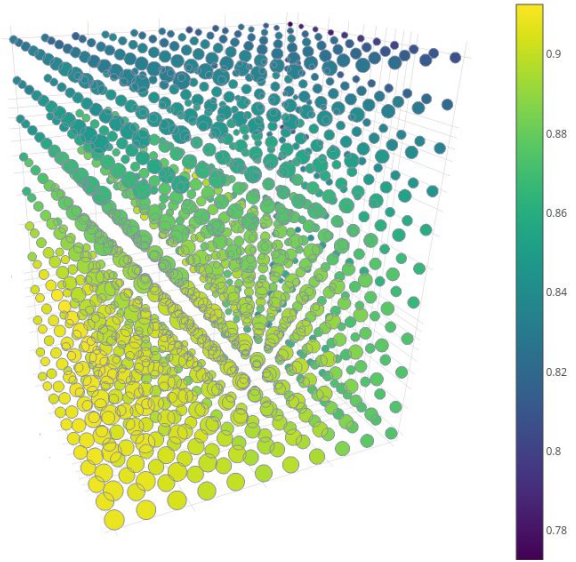


Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ **points** (parameter **combinations**)!



Grid search: For each parameter, we choose a finite number of possible values. We try each combination of the possible values across all parameters.

Practically, we evaluate the loss function in **each point of a grid** defined in the parameter space.

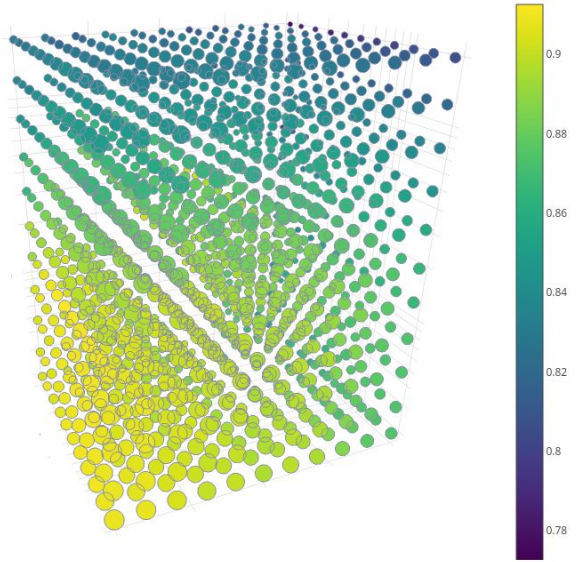
Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ **points** (parameter **combinations**)!

Any problem?



Linear regression - The least squares method

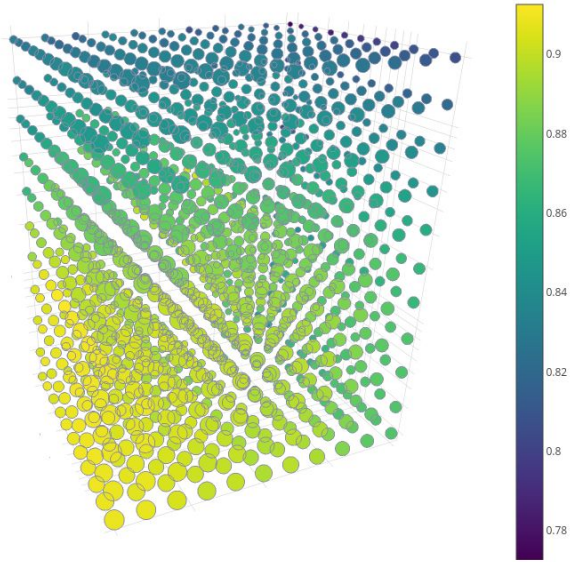
How do we find the optimal parameter(s) θ*?

Naive approach (1):

Grid search - Evaluate J(θ) in several θ **points** (parameter **combinations**)!

Problem: When our model will have more than one parameters, the parameter space will also become multidimensional:

the **number of parameter combinations** to evaluate **grows exponentially** with the number of dimensions!

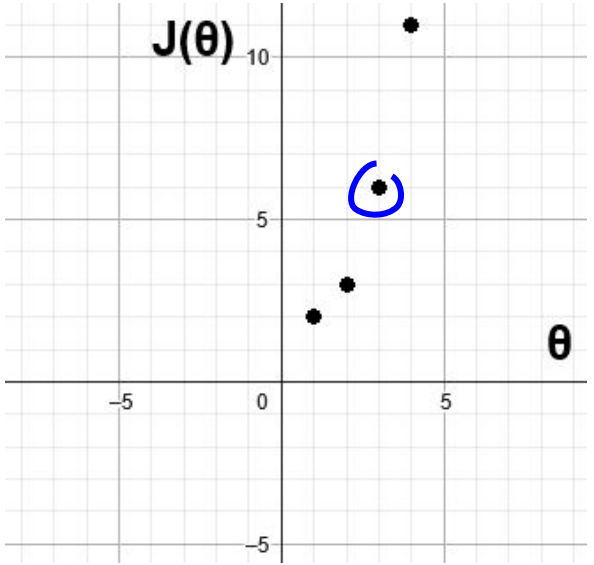


Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (2): Let's examine the **neighbors of point θ**! Let's move in the **direction where the loss decreases**!

Randomly selected starting point: θ = 3



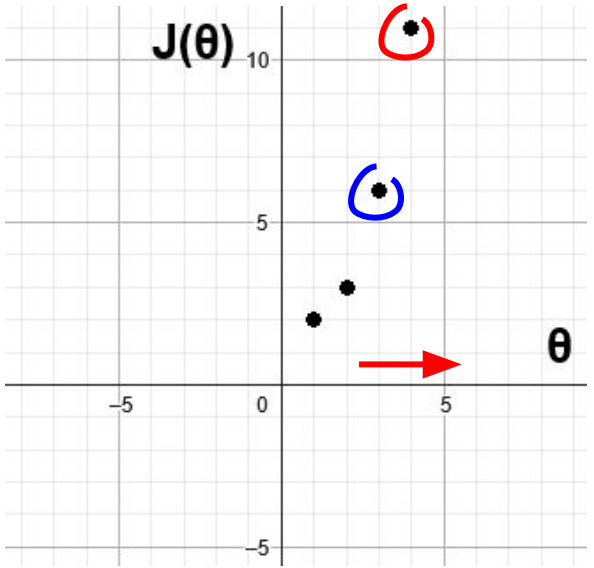
Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (2): Let's examine the **neighbors of point θ**! Let's move in the **direction where the loss decreases**!

Randomly selected starting point: θ = 3

Check θ = 4: J(4) > J(3) → wrong direction



Linear regression - The least squares method

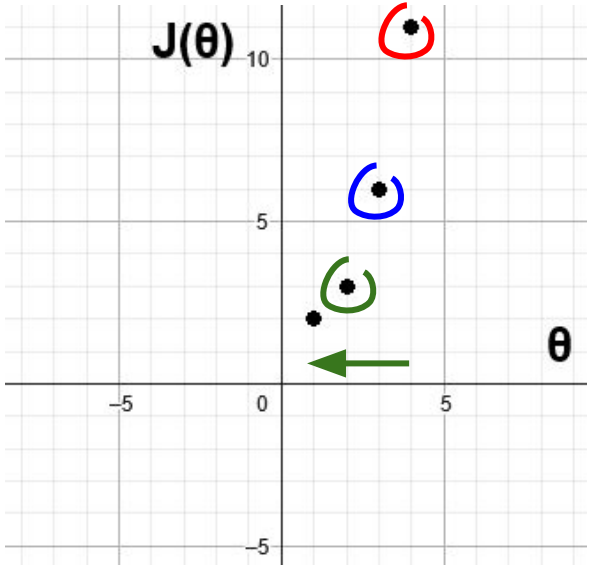
How do we find the optimal parameter(s) θ*?

Naive approach (2): Let's examine the **neighbors of point θ**! Let's move in the **direction where the loss decreases**!

Randomly selected starting point: θ = 3

Check θ = 4: J(4) > J(3) → wrong direction

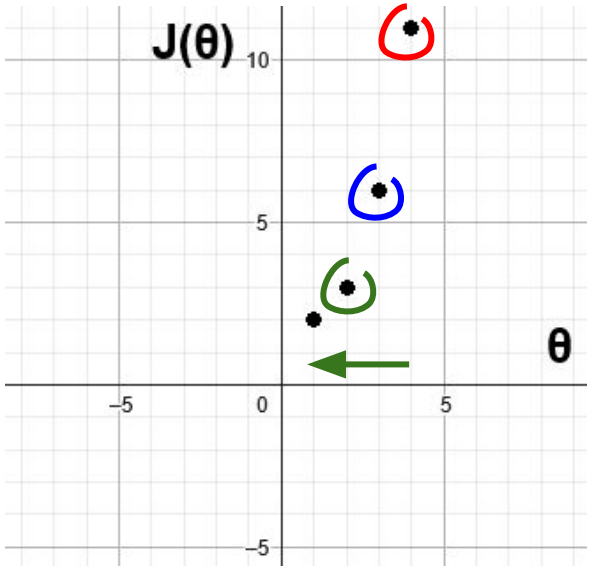
Check θ = 2: J(2) < J(3) → good direction



Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (2): Let's examine the **neighbors of point θ**! Let's move in the **direction where the loss decreases**!



This is a more efficient method than grid search.

However, as the number of parameters increases, the **number of directions** to be tested also **increases exponentially**...

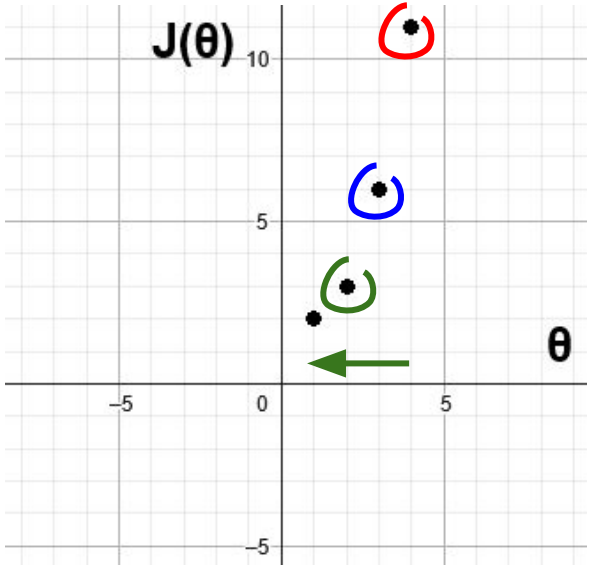
Linear regression - The least squares method

How do we find the optimal parameter(s) θ*?

Naive approach (2): Let's examine the **neighbors of point θ**! Let's move in the **direction where the loss decreases**!

There is something we haven't considered yet…

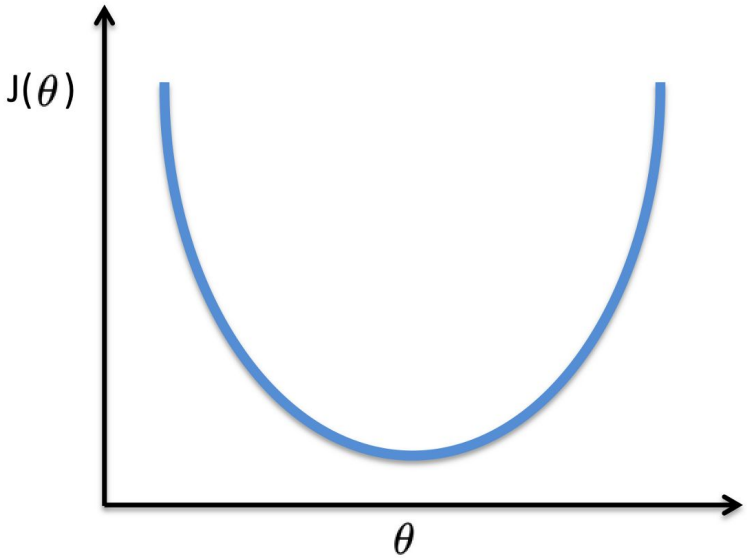
We know the formula for the loss function!



Linear regression - The least squares method

Loss function J is quadratic.

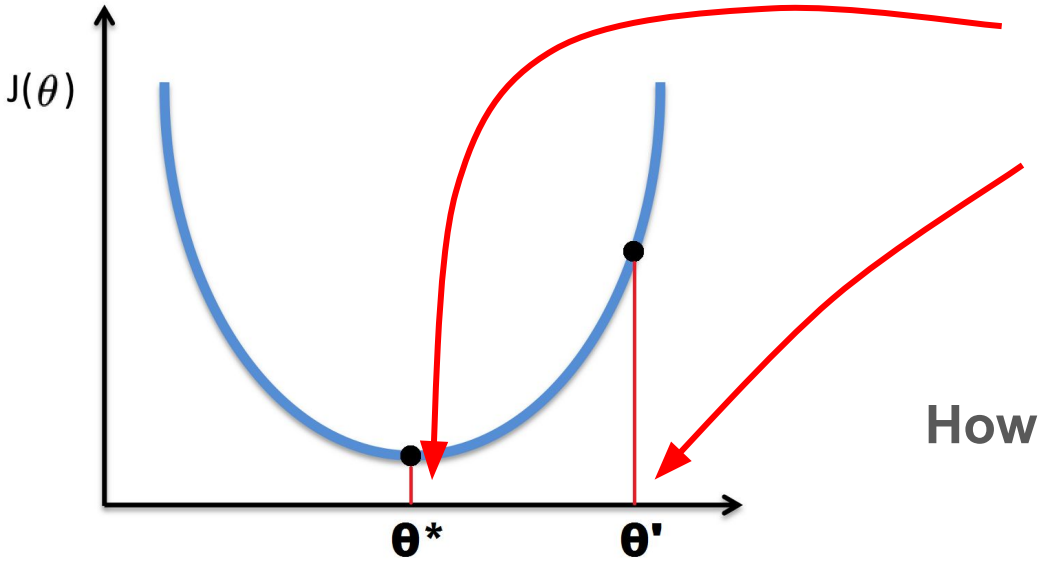
Since we only have a single **θ** parameter, now **J** is a parabola.



$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

Linear regression - The least squares method

Our goal: $\theta^* = \underset{\theta}{\operatorname{argmin}}\, J(\theta)$



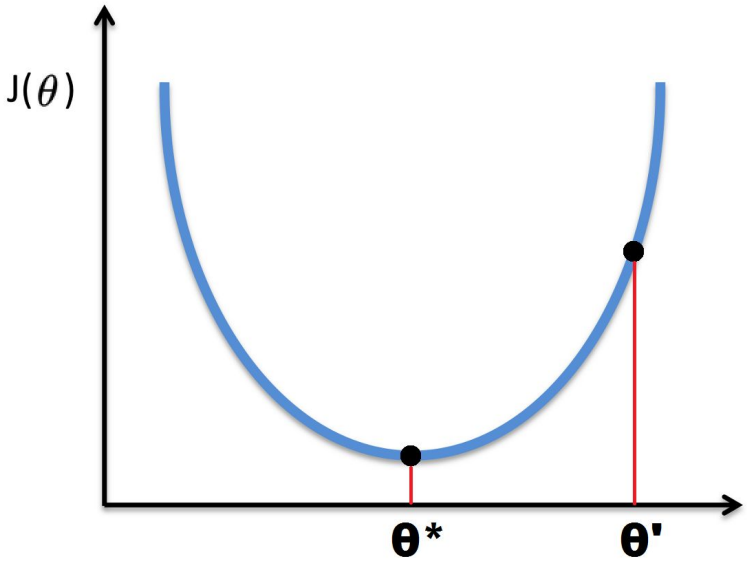
Optimal parameter value: θ*

Initial (actual) parameter value: θ'

How do we get from θ' to θ*?

Linear regression - The least squares method

How do we get from θ' to θ*?

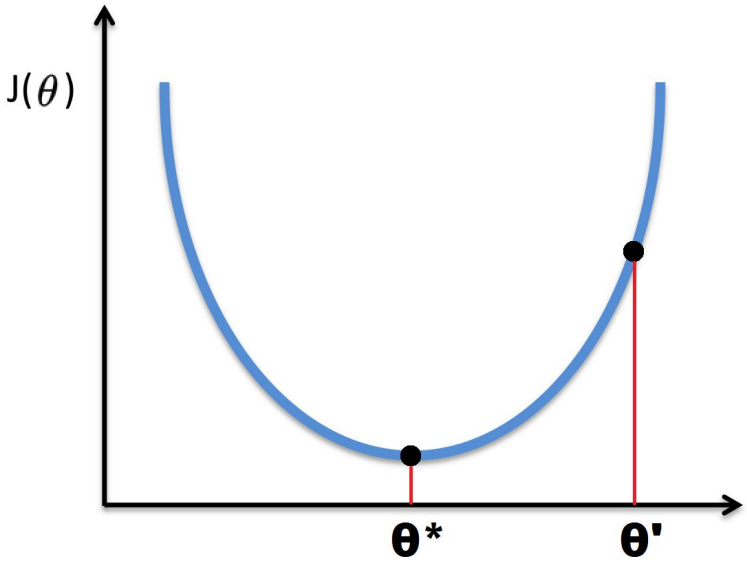


Let's see **which direction** the loss function slopes **most steeply downwards** at the current parameter value θ'!

What determines the slope of J at a given point θ'?

Linear regression - The least squares method

How do we get from θ' to θ*?



Let's see **which direction** the loss function slopes **most steeply downwards** at the current parameter value θ'!

What determines the slope of J at a given point θ'?

The derivative of J at point θ'.

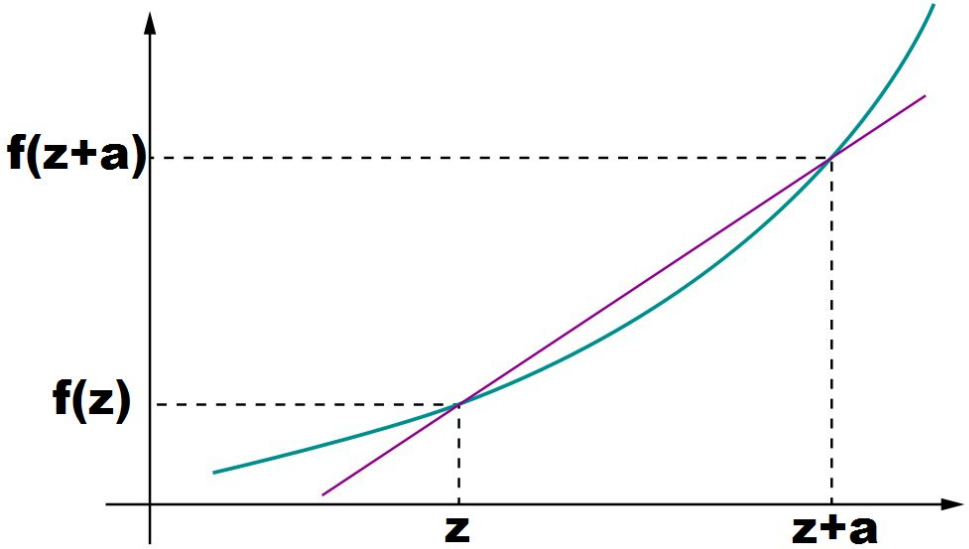
Linear regression - The least squares method

How do we define the derivative of a function?

Linear regression - The least squares method

How do we define the derivative of a function?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$



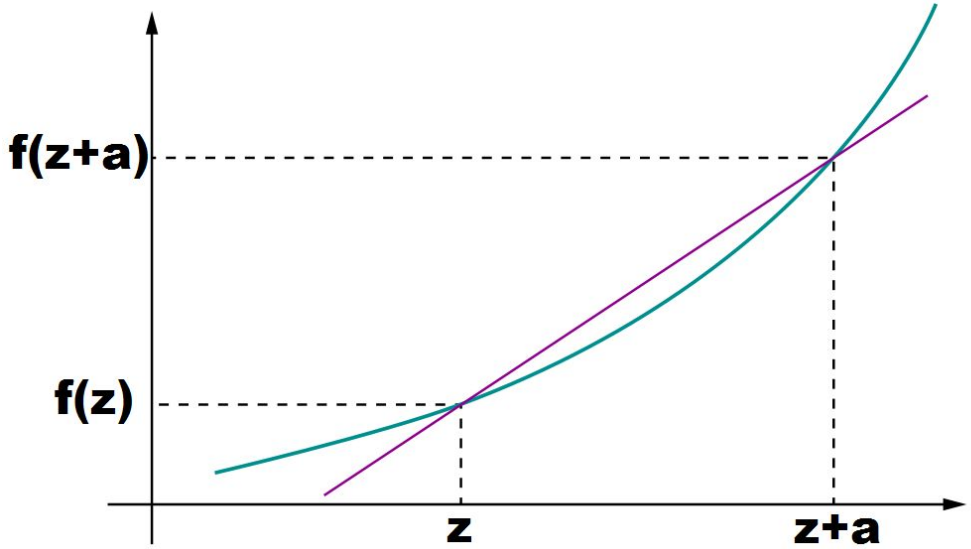
Linear regression - The least squares method

How do we define the derivative of a function?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$

The difference quotient:

The **slope of the chord** connecting the two points on the function curve: (z, f(z)) and (z+a, f(z+a))

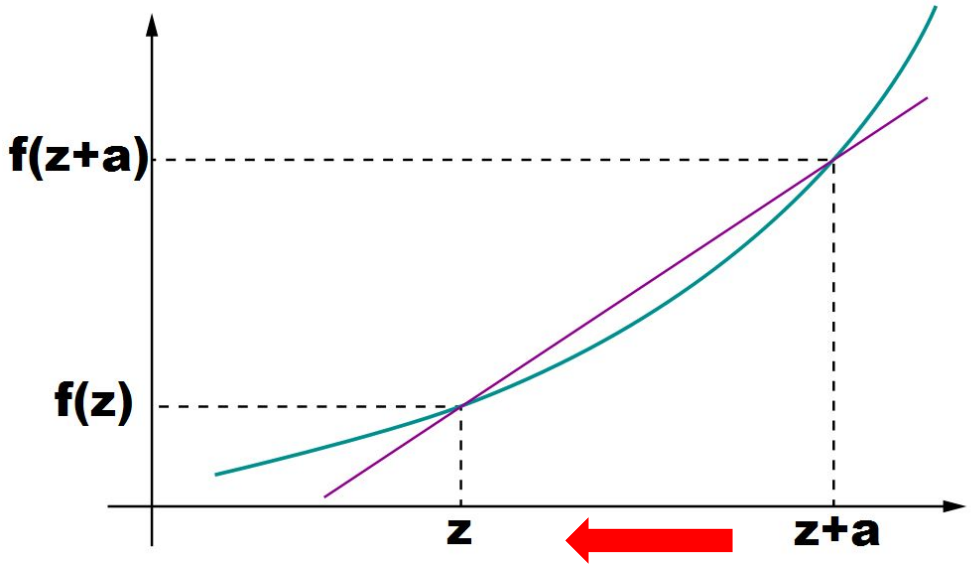


Linear regression - The least squares method

How do we define the derivative of a function?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$

The derivative of a function at point **z** is **the limit of the slope of the chord** when **a** approaches zero…
… if this limit exists and is finite.



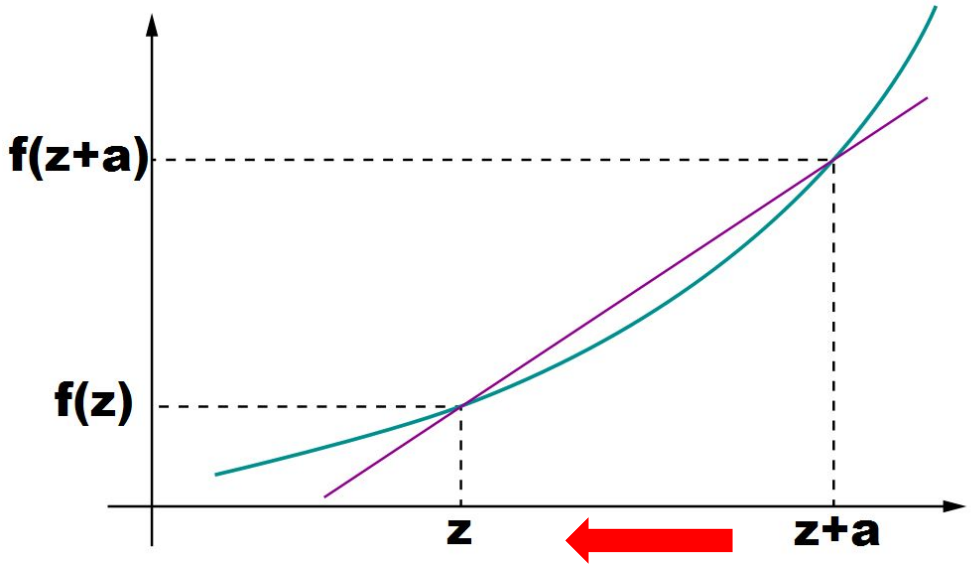
Linear regression - The least squares method

How do we define the derivative of a function?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$

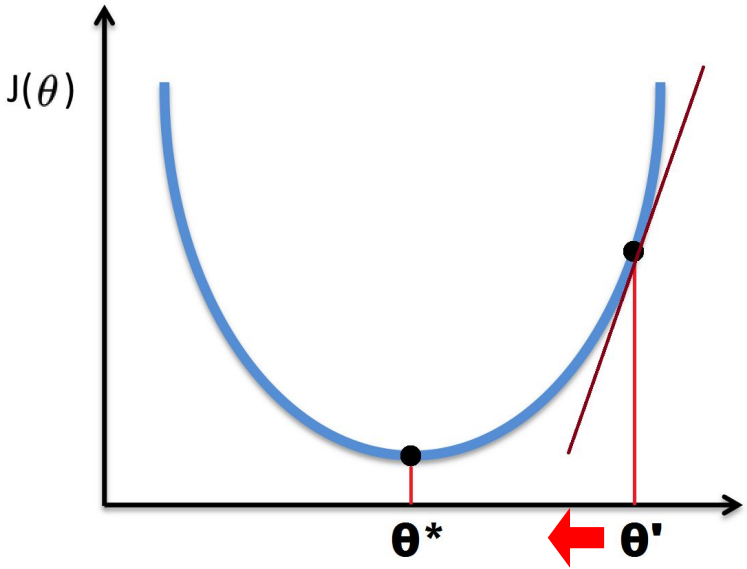
The derivative of a function at point **z** is **the limit of the slope of the chord** when **a** approaches zero…
… if this limit exists and is finite.

This limit is equal to the slope of the tangent to the curve at point **z**.



Linear regression - The least squares method

How do we know what direction to take in order to reduce the loss?



Let's move from **θ'** in the direction where the loss function curve slopes downwards.

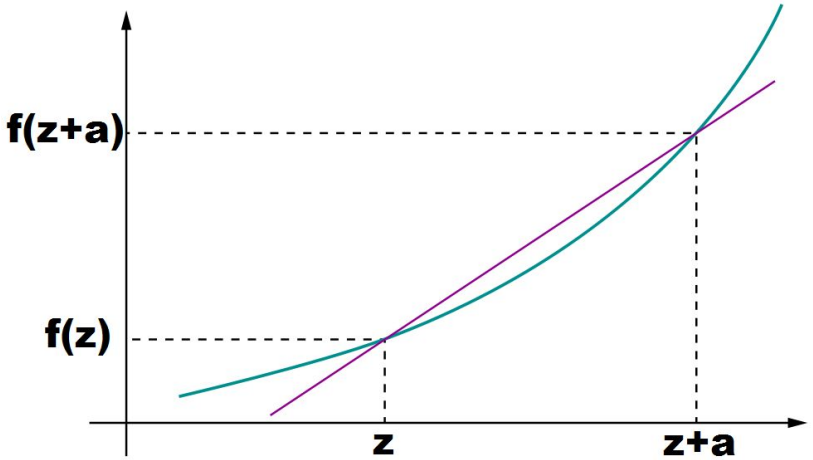
To do this, we need to calculate the derivative of the loss function J.

Linear regression - The least squares method

We need to calculate the derivative of the loss function J.

Do we have to use the difference quotient formula?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$

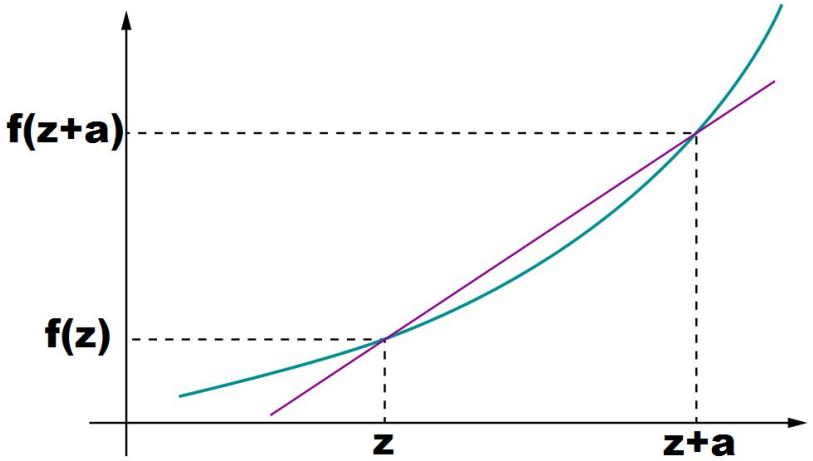


Linear regression - The least squares method

We need to calculate the derivative of the loss function J.

Do we have to use the difference quotient formula?

$$f'(z) = \lim_{a \to 0} \frac{f(z+a) - f(z)}{a}$$



Not necessarily!

We have higher level tools to compute the derivative.

Linear regression - The least squares method

The rules of symbolic differentiation

$$\text{If } f(x) = x^n \text{ then } \frac{df(x)}{dx} = nx^{n-1}$$

$$\text{If } f(x) = k \text{ then } \frac{df(x)}{dx} = 0$$

$$\text{If } f(x) = g(x) + h(x) \text{ then } \frac{df(x)}{dx} = \frac{dg(x)}{dx} + \frac{dh(x)}{dx}$$

$$\text{If } f(x) = g(x)h(x) \text{ then } \frac{df(x)}{dx} = \frac{dg(x)}{dx}h(x) + g(x)\frac{dh(x)}{dx}$$

A list of rules:

https://homepage.cs.uiowa.edu/~stroyan/CTLC3rdEd/3rdCTLCText/Chapters/ch6.pdf

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta} J(\theta) = \ ?$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

Remember: x and **y** are known values from the training set, only **θ** is unknown.

$$\frac{\partial}{\partial \theta} J(\theta) = \ ?$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta} J(\theta) = \ ?$$

Derivative of the composition of two differentiable functions
(chain rule of calculus):
f(g(z))' = f'(g(z)) · g'(z)

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(z))' = f'(g(z)) · g'(z)

Leibniz notation for partial differentiation:
J is differentiated with respect to **θ**. Here we only have one variable (**θ**), so there is no question as to what we are differentiating with respect to...

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\underbrace{\theta x^{(j)} - y^{(j)}}_{g(\theta) = \theta x^{(j)} - y^{(j)}})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(θ))' = f'(g(θ)) · g'(θ)

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\underbrace{\theta x^{(j)} - y^{(j)}}_{g(\theta) = \theta x^{(j)} - y^{(j)}})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(θ))' = f'(g(θ)) · g'(θ)

$$g'(\theta) = x^{(j)}$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

$$f(u) = \frac{1}{2m}\sum_{j=1}^{m} u^2$$

$$u := g(\theta)$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(θ))' = f'(g(θ)) · g'(θ)

$$g(\theta) = \theta x^{(j)} - y^{(j)}$$

$$g'(\theta) = x^{(j)}$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

$$f(u) = \frac{1}{2m}\sum_{j=1}^{m} u^2$$

$$u := g(\theta)$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(θ))' = f'(g(θ)) · g'(θ)

$$g(\theta) = \theta x^{(j)} - y^{(j)}$$

$$g'(\theta) = x^{(j)}$$

$$f'(u) = \frac{1}{2m}\sum_{j=1}^{m} 2 \cdot u = \frac{1}{m}\sum_{j=1}^{m} u$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \ ?$$

Derivative of the composition of two differentiable functions (chain rule of calculus):
f(g(θ))' = f'(g(θ)) · g'(θ)

$$g(\theta) = \theta x^{(j)} - y^{(j)} \qquad g'(\theta) = x^{(j)}$$

$$f(u) = \frac{1}{2m}\sum_{j=1}^{m}u^2 \qquad f'(u) = \frac{1}{2m}\sum_{j=1}^{m}2\cdot u = \frac{1}{m}\sum_{j=1}^{m}u$$

$$u := g(\theta)$$

$$f(g(\theta))' = f'(u)\cdot g'(\theta) = \frac{1}{m}\sum_{j=1}^{m}u\cdot x^{(j)} = \frac{1}{m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})x^{(j)}$$

Linear regression - The least squares method

The derivative of the loss function:

$$J(\theta) = \frac{1}{2m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})^2$$

$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{2m}\sum_{j=1}^{m} 2\cdot(\theta x^{(j)} - y^{(j)})\cdot(\theta x^{(j)} - y^{(j)})' =$$

$$= \frac{1}{m}\sum_{j=1}^{m}(\theta x^{(j)} - y^{(j)})x^{(j)}$$

Linear regression - The least squares method

Gradient descent

We repeatedly step with **θ** in the direction where
the slope of the loss function is greatest at the current **θ** parameter value.

$$
\begin{aligned}
&\text{repeat until convergence} \quad \{ \\
&\quad grad := \frac{\partial}{\partial \theta} J(\theta) \\
&\quad \theta := \theta - \alpha \cdot \ grad \\
&\}
\end{aligned}
$$

Linear regression - The least squares method

Gradient descent

We repeatedly step with **θ** in the direction where
the slope of the loss function is greatest at the current **θ** parameter value.

$$\text{repeat until convergence} \quad \{$$

$$grad := \frac{\partial}{\partial \theta} J(\theta)$$

$$\theta := \theta - \alpha \cdot \ grad$$

$$\}$$

$$\frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m} \sum_{j=1}^{m} (\theta x^{(j)} - y^{(j)}) x^{(j)}$$

Linear regression - The least squares method

Gradient descent

We repeatedly step with **θ** in the direction where
the slope of the loss function is greatest at the current **θ** parameter value.

$$\text{repeat until convergence} \quad \{$$

$$grad := \frac{\partial}{\partial \theta} J(\theta)$$

$$\theta := \theta - \alpha \cdot \ grad$$

$$\}$$

Gradient of J: Vector pointing in the direction of the maximum increase of **J**; its elements are the partial derivatives of **J** at a given point
(here still only single-valued)

alpha: the learning rate;
the size of the steps can be scaled with it

Linear regression - The least squares method

Gradient descent

We repeatedly step with **θ** in the direction where
the slope of the loss function is greatest at the current **θ** parameter value.

$$\text{repeat until convergence} \quad \{$$

$$grad := \frac{\partial}{\partial \theta} J(\theta)$$

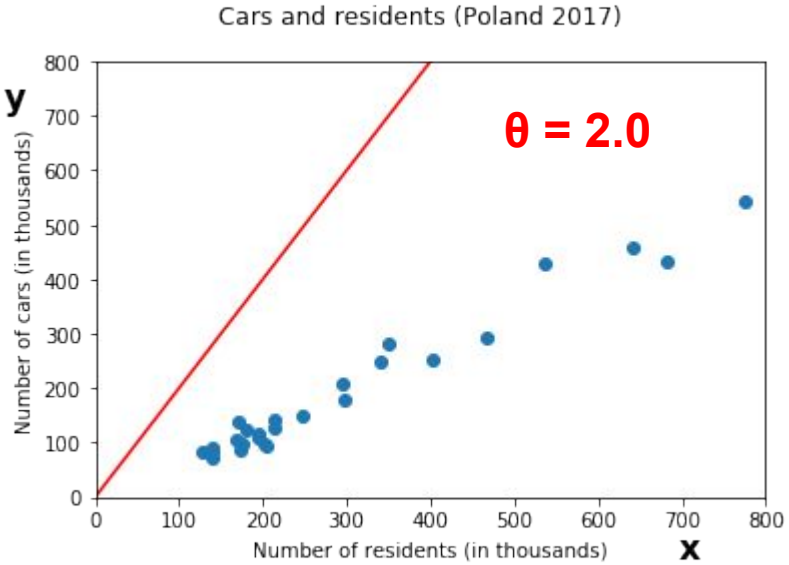
$$\theta := \theta - \alpha \cdot \ grad$$

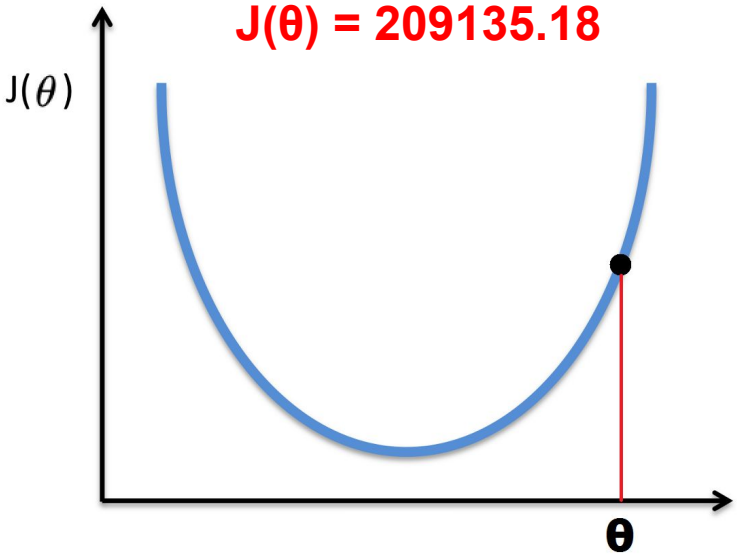
$$\}$$

We **subtract** the gradient from the current parameter value, as we are looking for the steepest descent.

Linear regression - The least squares method

Applying gradient descent, T = 0 (before taking the first step)

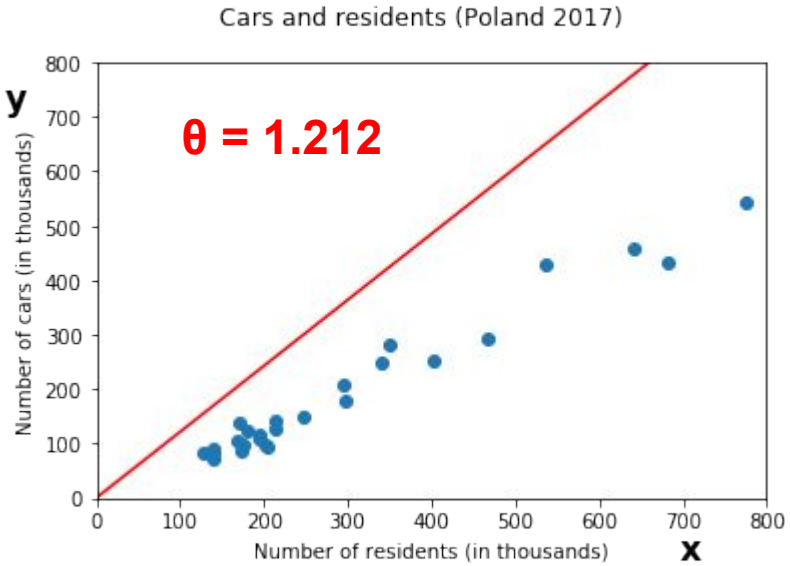


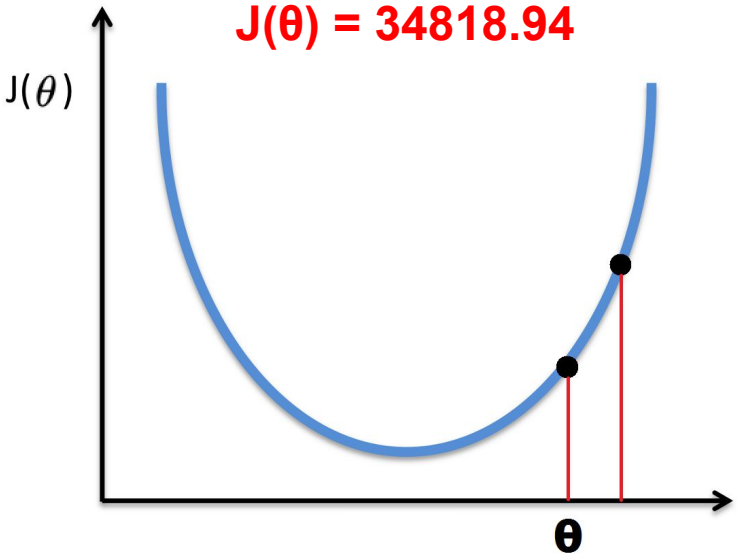


We can choose the initial parameter value (**θ**) randomly.

Linear regression - The least squares method

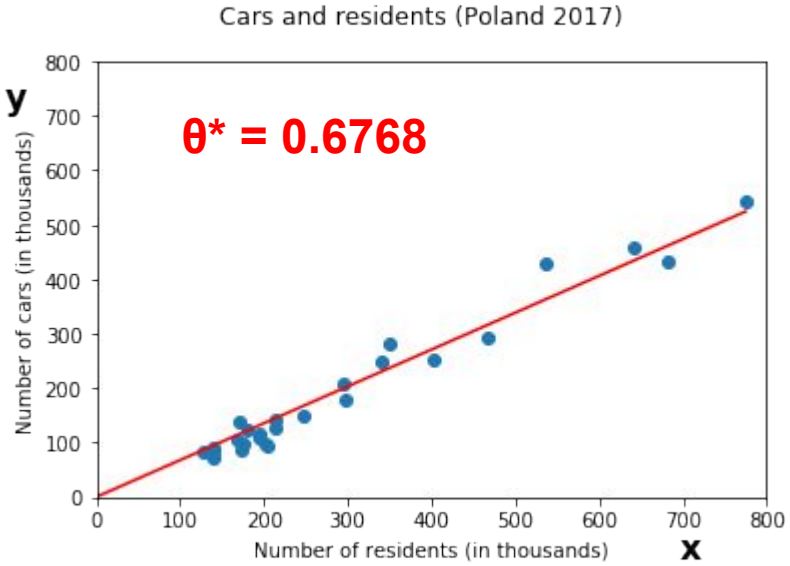
Applying gradient descent, T = 1

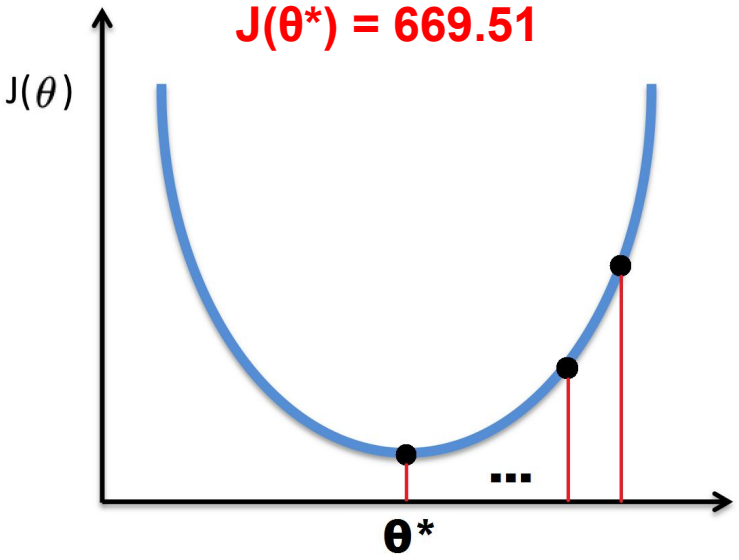




Linear regression - The least squares method

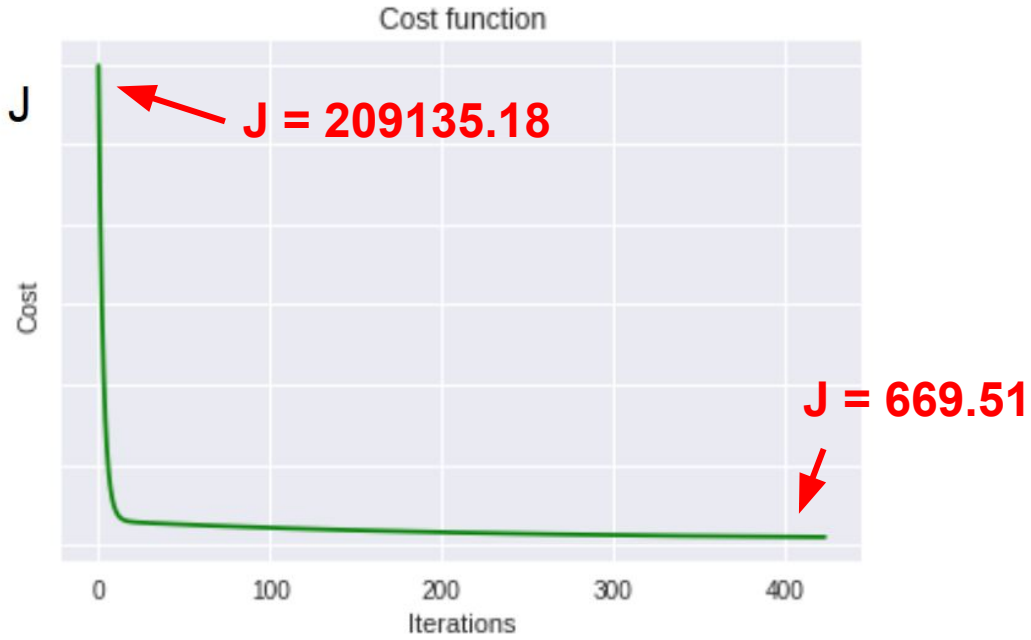
Applying gradient descent, T = < many >





Linear regression - The least squares method

Changes in the loss value during the steps of the gradient descent

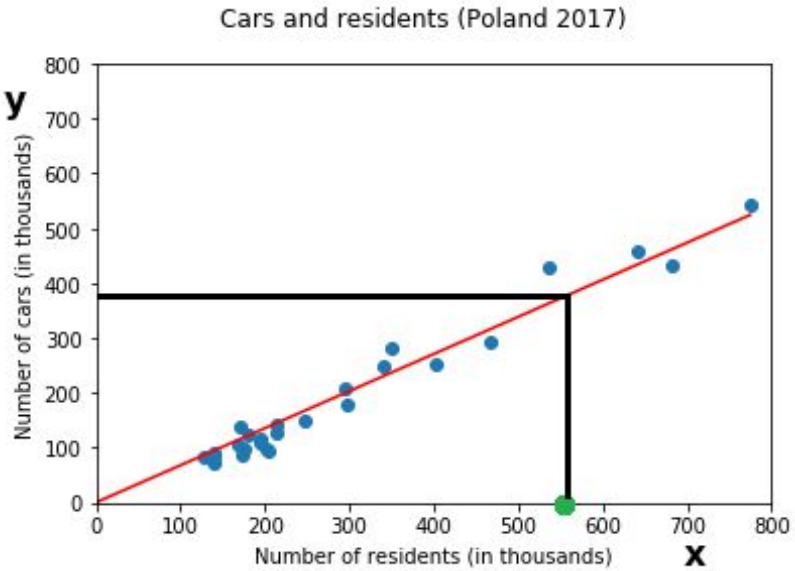


Linear regression

What have we achieved?

We trained a simple (linear) regression model.

We will be able to estimate labels for new, unlabeled examples.



Linear regression

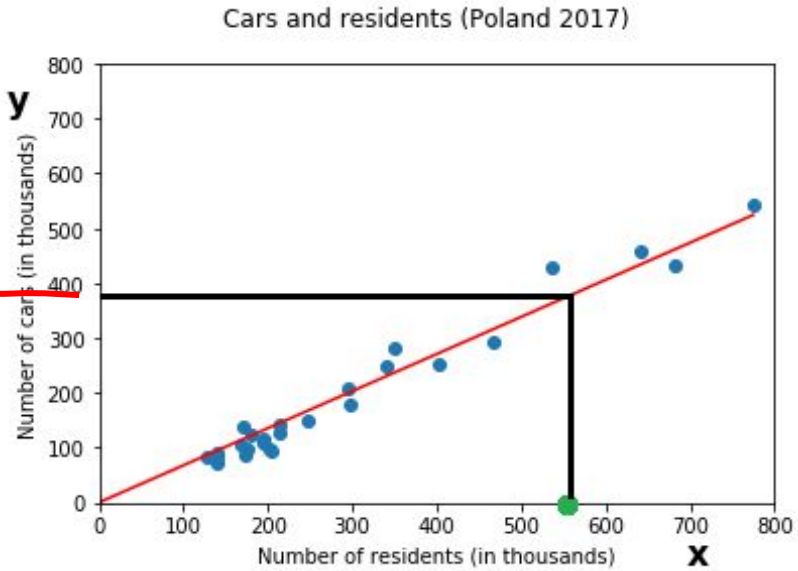
What have we achieved?

We trained a simple (linear) regression model.

We will be able to estimate labels for new, unlabeled examples.

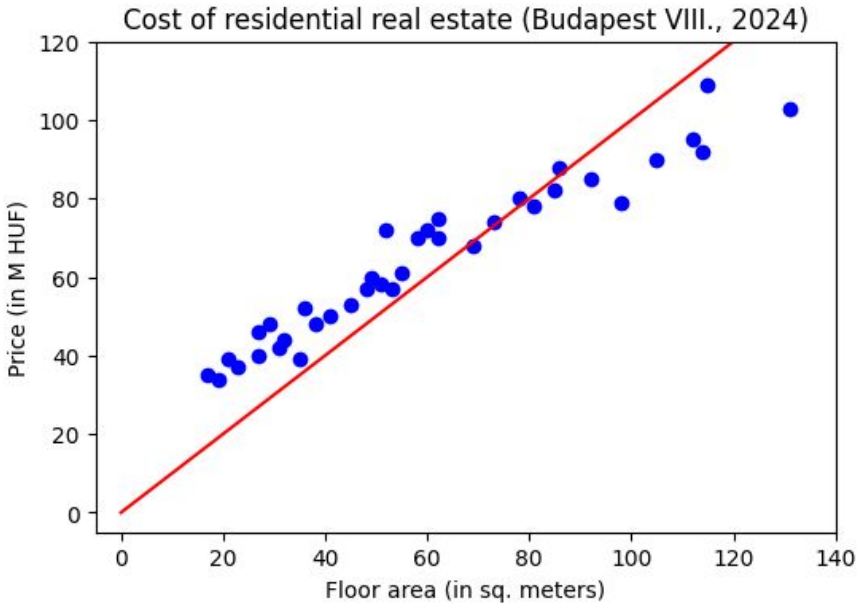
For example, for a **city with a population of 550 000**, we estimate

0.6768 * 550 000 = **372 240 cars.**



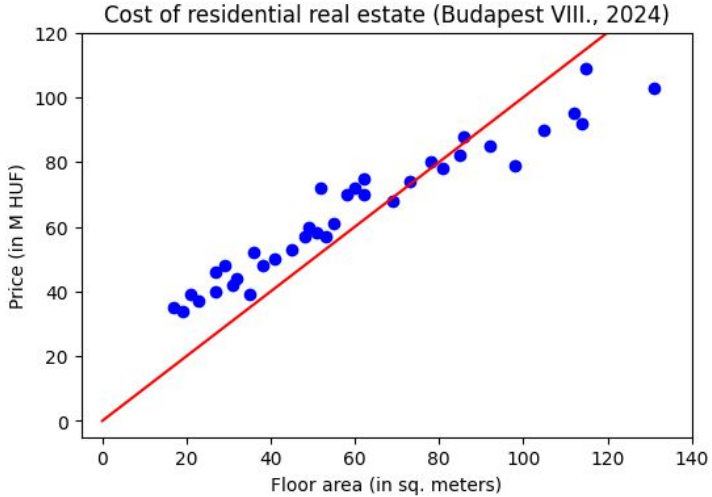
Linear regression

What's missing?



Linear regression

What's missing?



New dataset: Apartment prices as a function of floor space.

Problem: Smaller apartments are in higher demand, so their price per square meter rate is higher.

→ **Does not fit well with the straight line passing through the origin...**

Linear regression

What's missing?

Our model so far has been too limited. The hypothesis function was a straight line that had to pass through the origin...

In addition to the slope, we also introduce a "constant" (bias / intercept) parameter.

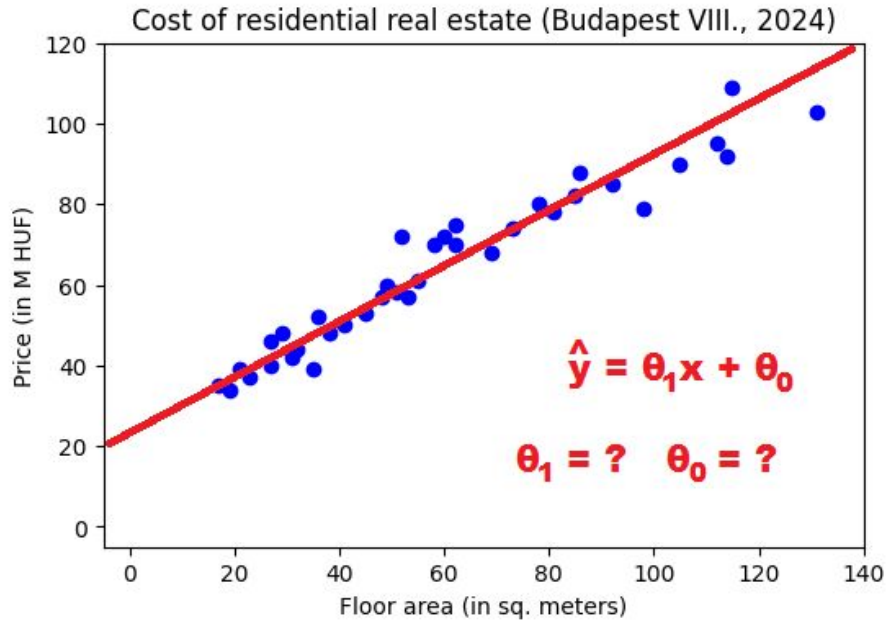
Former hypothesis: $h(x) = \theta x = \hat{y} \approx y$

New hypothesis: $h(x) = \theta_1 x + \theta_0 = \hat{y} \approx y$

Linear regression

The new hypothesis function: $y \approx \hat{y} = h(x) = \theta_1 x + \theta_0$

$\theta_1, \theta_0 \in \mathbb{R}$



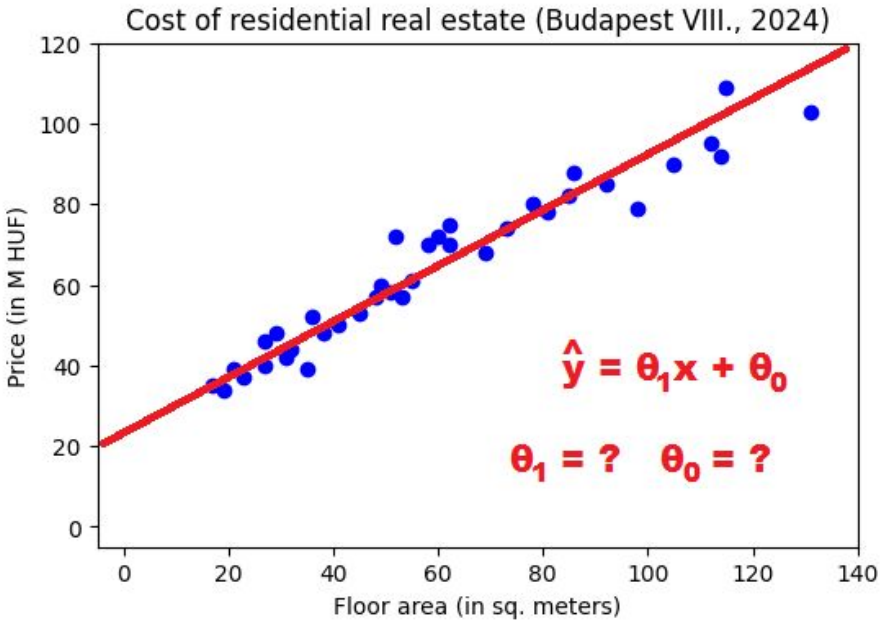
Linear regression

The new hypothesis function: $y \approx \hat{y} = h(x) = \theta_1 x + \theta_0$

$$\theta_1, \theta_0 \in \mathbb{R}$$

θ_1 the slope of the line

θ_0 is the value where the line intersects the y-axis (the bias / intercept).



Linear regression - The least squares method

The loss function is still the **Mean Squared Error** (MSE):

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{m} (\overbrace{h_\theta(x^{(j)})}^{\hat{y}^{(j)}} - y^{(j)})^2$$

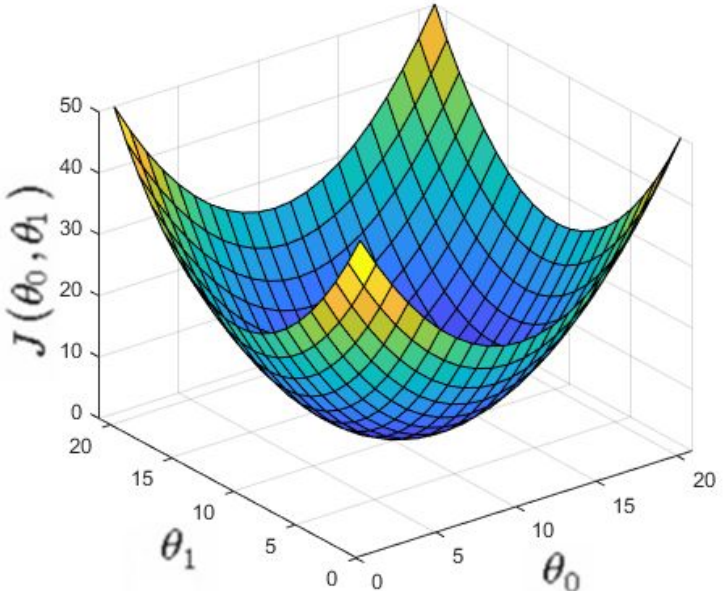
However, the hypothesis function has changed.

What will the loss function graph look like?

Linear regression - The least squares method

Loss function J is still quadratic.

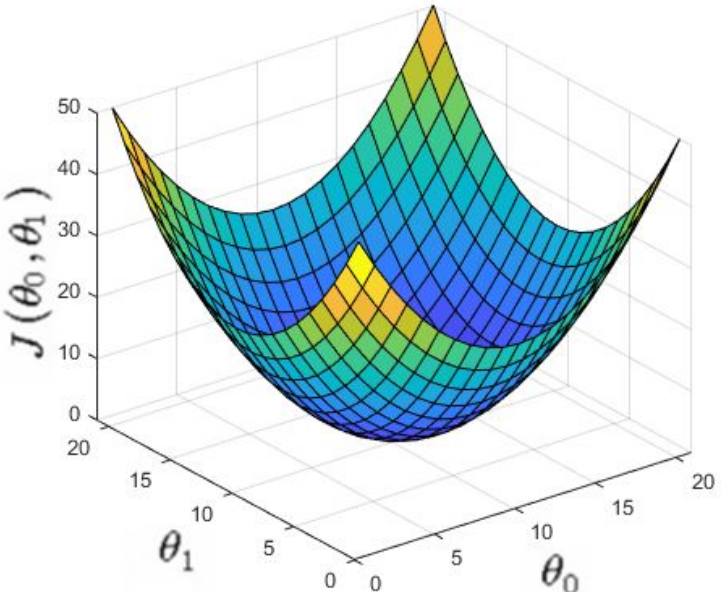
Since we have **two parameters** now, it is an elliptic paraboloid.



$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^{m} (\overbrace{\theta_1 x^{(j)} + \theta_0}^{\hat{y}^{(j)}} - y^{(j)})^2$$

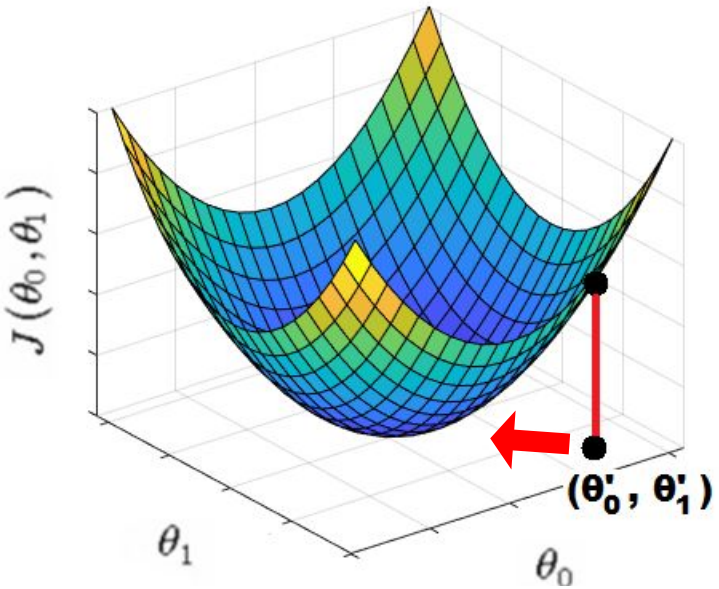
Linear regression - The least squares method

How do we know what direction to take in order to reduce the loss?



Linear regression - The least squares method

How do we know what direction to take in order to reduce the loss?



Let's use the gradient method:
Let's move from **θ'** in the direction where the loss function curve slopes downwards!

This direction will be given by the **gradient vector at θ'**.
The elements of the vector are the **partial derivatives** of the loss function **J** with respect to each parameter.

Linear regression - The least squares method

The partial derivatives of the loss function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \ ?$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \ ?$$

Linear regression - The least squares method

The partial derivatives of the loss function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2$$

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)}) \cdot x^{(j)}$$

Linear regression - The least squares method

The partial derivatives of the loss function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})^2$$

In **partial differentiation**, we differentiate the function with respect to **one variable**. In this case, **we treat the other variables as constants**.

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})$$

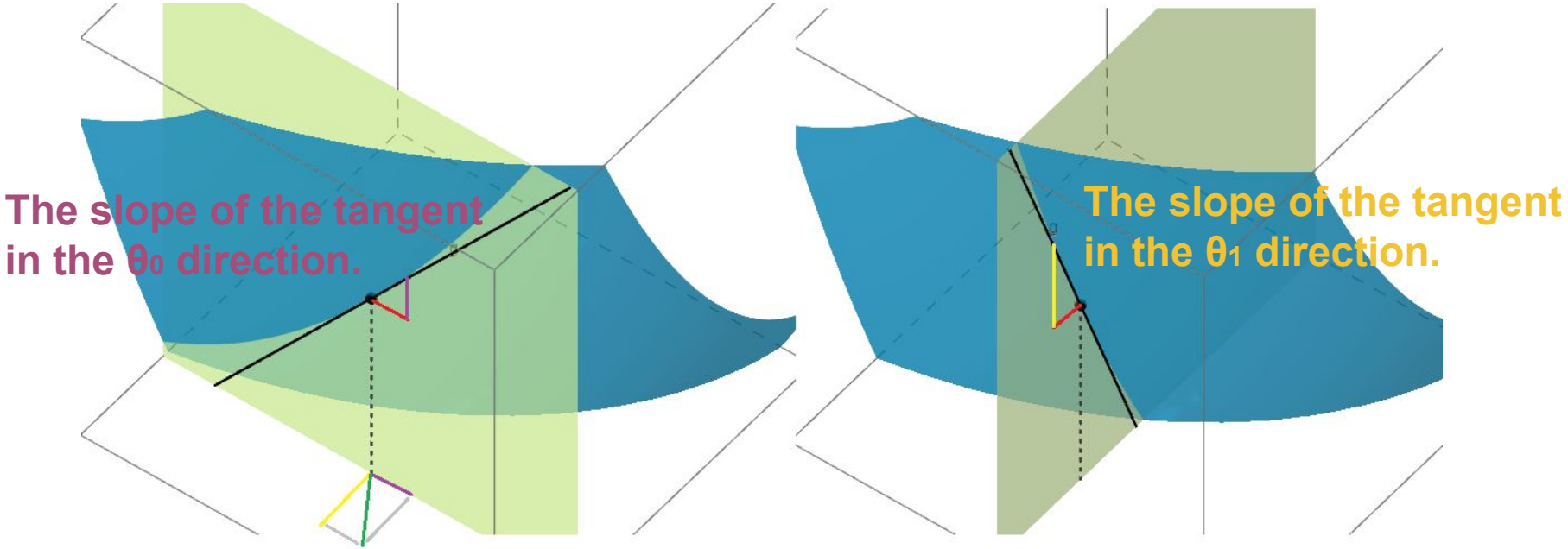
The slope of the tangent in the **θ_0** direction.

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)}) \cdot x^{(j)}$$

The slope of the tangent in the **θ_1** direction.

Linear regression - The least squares method

The partial derivatives of the loss function

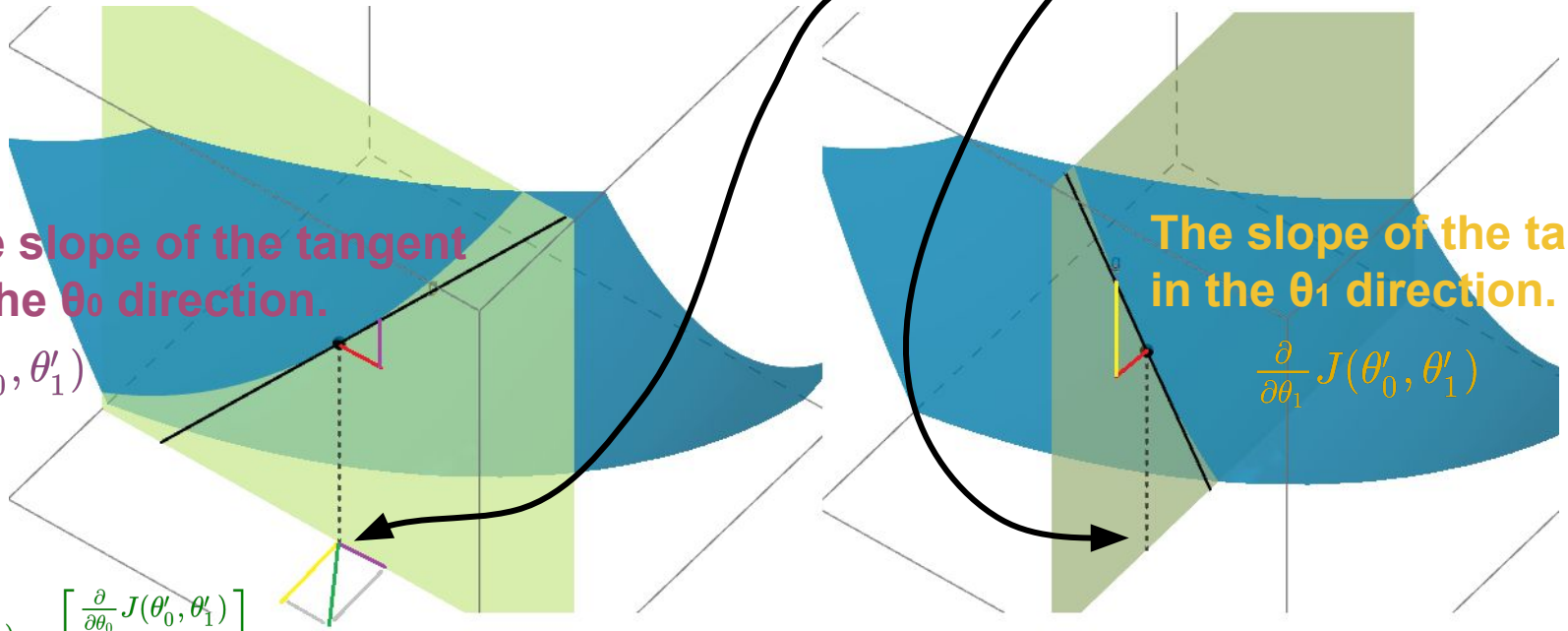


The gradient vector: the direction of the steepest increase

Linear regression - The least squares method

The partial derivatives of the loss function

The point of the actual parameters (θ'_0, θ'_1)



The slope of the tangent in the θ₀ direction.

$$\frac{\partial}{\partial\theta_0} J(\theta'_0, \theta'_1)$$

The slope of the tangent in the θ₁ direction.

$$\frac{\partial}{\partial\theta_1} J(\theta'_0, \theta'_1)$$

$$\nabla J(\theta'_0, \theta'_1) = \begin{bmatrix} \frac{\partial}{\partial\theta_0} J(\theta'_0, \theta'_1) \\ \frac{\partial}{\partial\theta_1} J(\theta'_0, \theta'_1) \end{bmatrix}$$

The gradient vector: the direction of the steepest increase

Linear regression - The least squares method

The gradient descent algorithm with two parameters:

$$
\begin{aligned}
&\text{repeat until convergence} \quad \{ \\
&\quad grad_0 := \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\quad grad_1 := \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\
&\quad \theta_0 := \theta_0 - \alpha \cdot \ grad_0 \\
&\quad \theta_1 := \theta_1 - \alpha \cdot \ grad_1 \\
&\}
\end{aligned}
$$

Linear regression - The least squares method

The gradient descent algorithm with two parameters:

repeat until convergence {

$$grad_0 := \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$grad_1 := \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \theta_0 - \alpha \cdot grad_0$$

$$\theta_1 := \theta_1 - \alpha \cdot grad_1$$

}

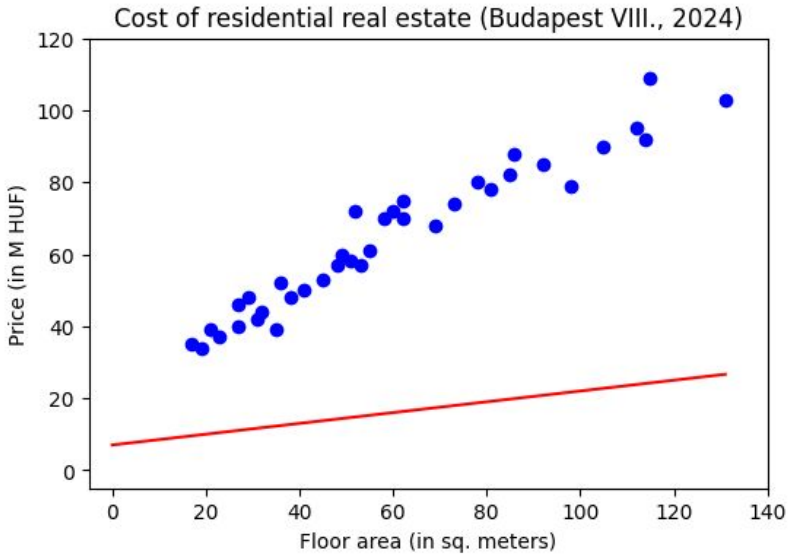
$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)})$$

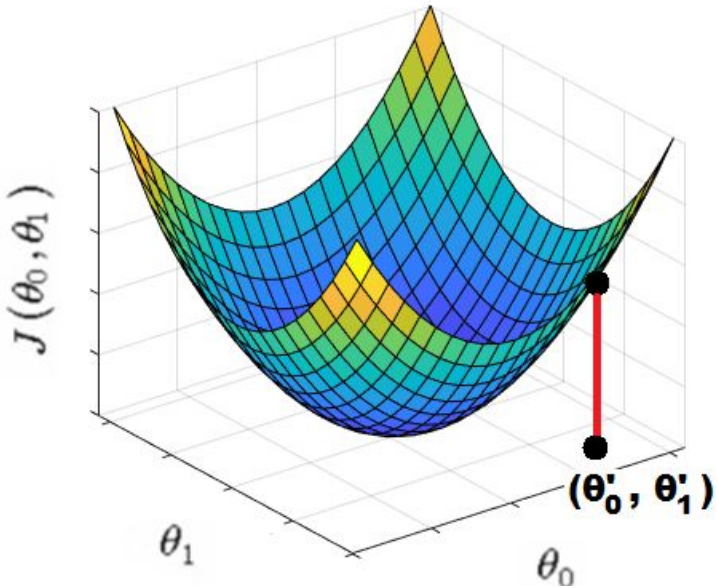
$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{j=1}^{m} (\theta_1 x^{(j)} + \theta_0 - y^{(j)}) \cdot x^{(j)}$$

alpha: the learning rate;
the size of the steps can be scaled with it

Linear regression - The least squares method

Applying gradient descent, T = 0 (before taking the first step)

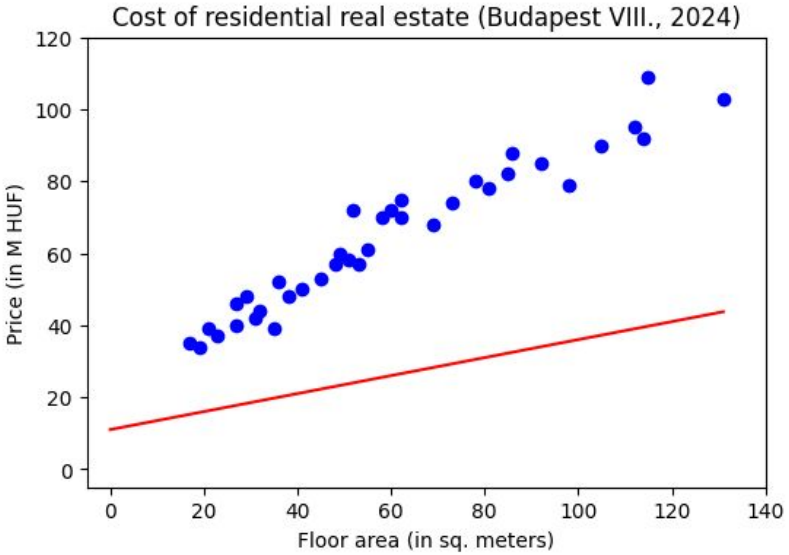


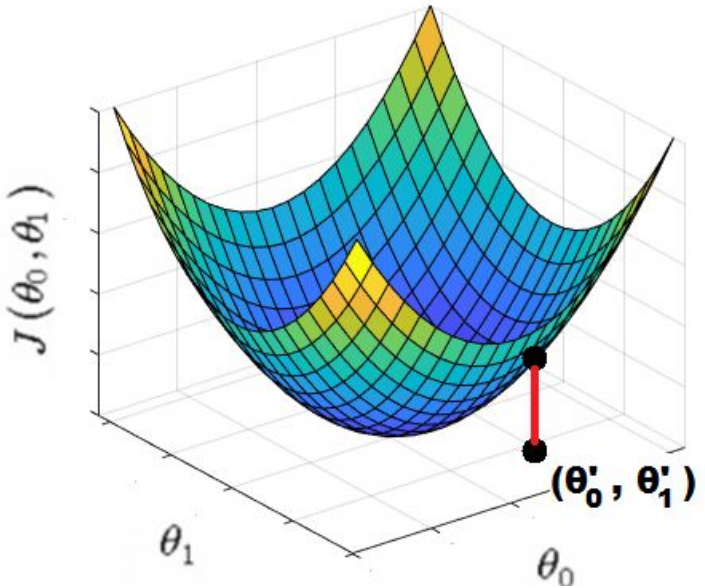


We can choose the initial parameters (θ_0, θ_1) randomly.

Linear regression - The least squares method

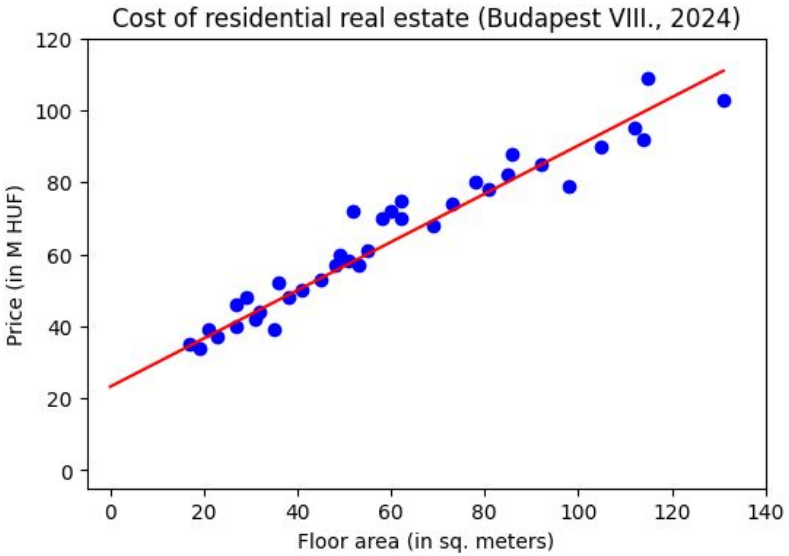
Applying gradient descent, T = 1

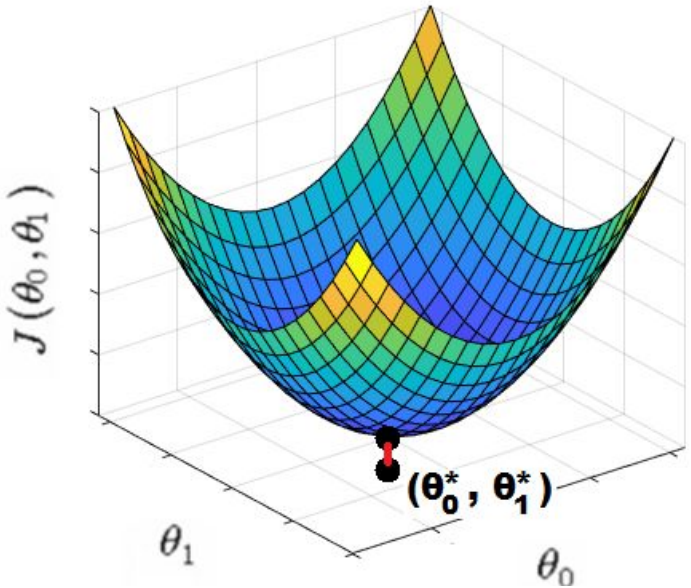




Linear regression - The least squares method

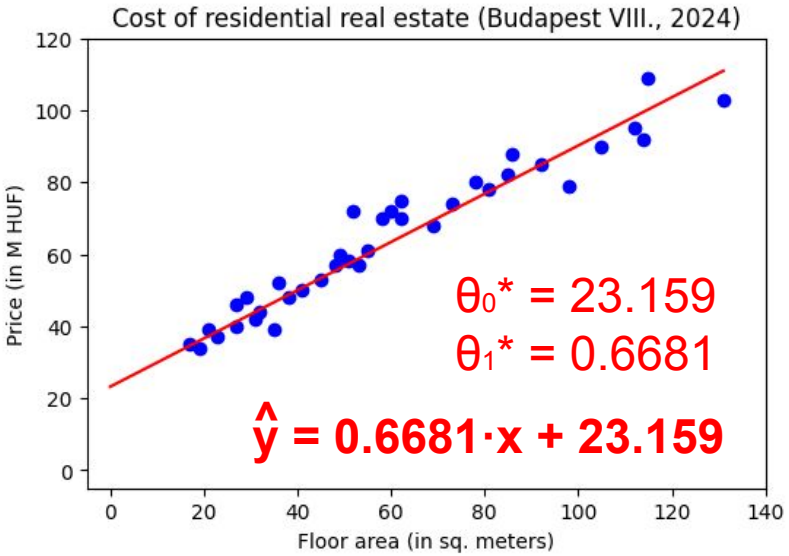
Applying gradient descent, T = < many >

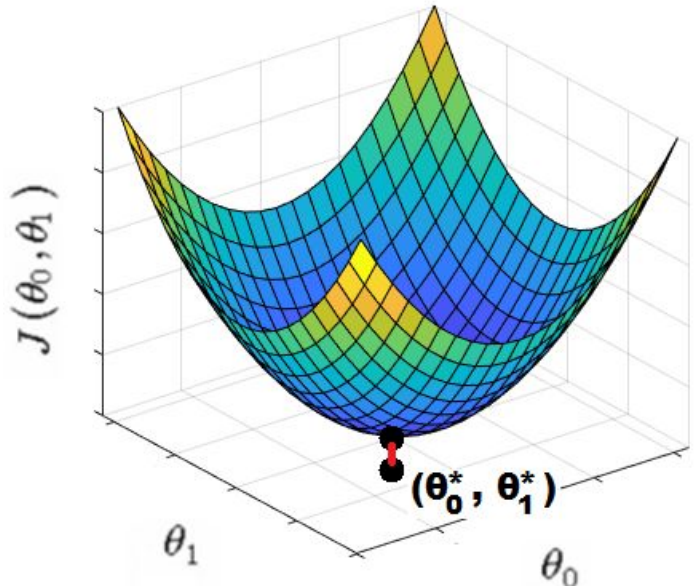




Linear regression - The least squares method

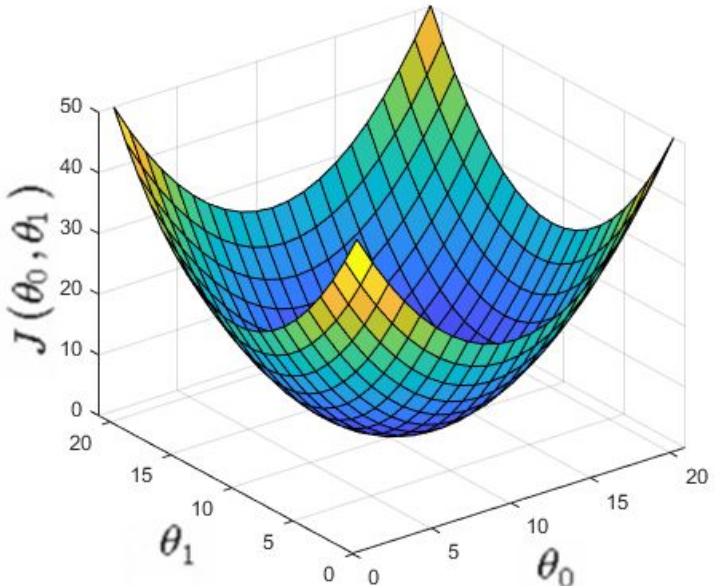
Applying gradient descent, T = < many >





Linear regression - The least squares method

Is it guaranteed that we will find the optimal solution
(minimum MSE loss) **for linear regression using gradient descent?**



Linear regression - The least squares method

Is it guaranteed that we will find the optimal solution
(minimum MSE loss) **for linear regression using gradient descent?**

Yes, if the step size (alpha)
is sufficiently small.

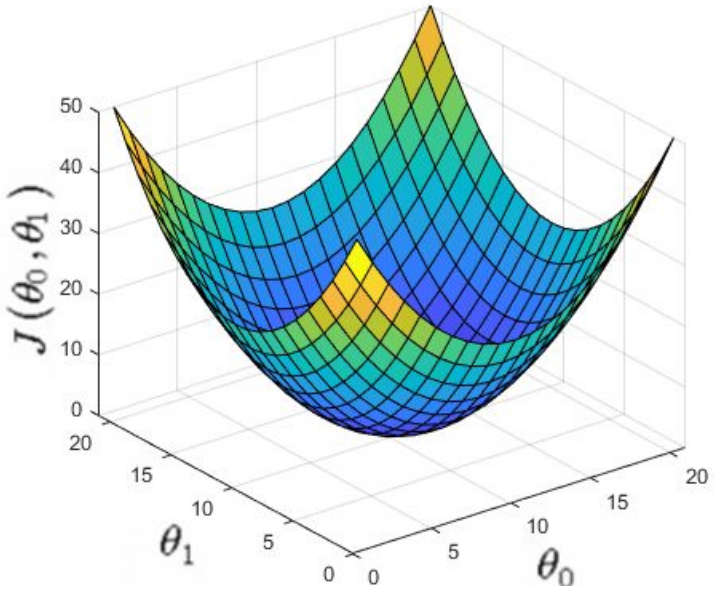


Image sources: spektr @ scicomp.stackexchange.com

Linear regression - The least squares method

Is it guaranteed that we will find the optimal solution
(minimum MSE loss) **for any function using gradient descent?**

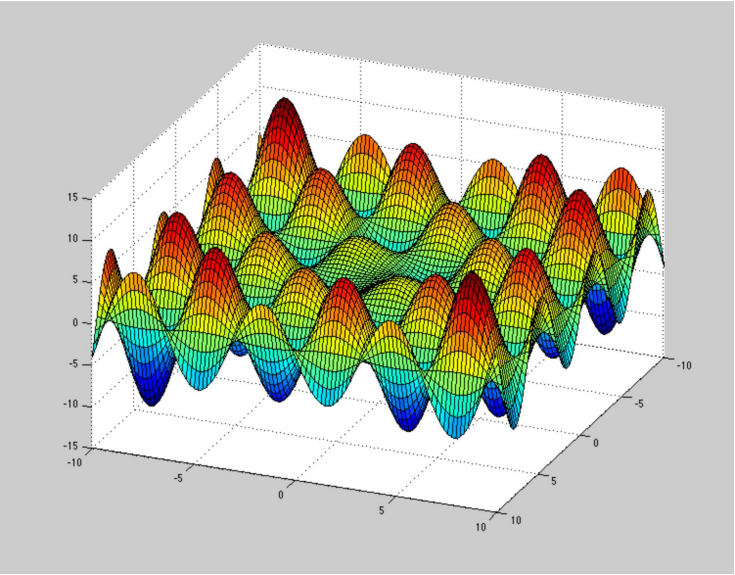


Image sources: spektr @ scicomp.stackexchange.com

Linear regression - The least squares method

Is it guaranteed that we will find the optimal solution
(minimum MSE loss) **for any function using gradient descent?**

No. We can reach
one of the **local minimum points**,
but if the loss function is **not convex**,
then it is **not guaranteed**
that this will be the **global minimum**.

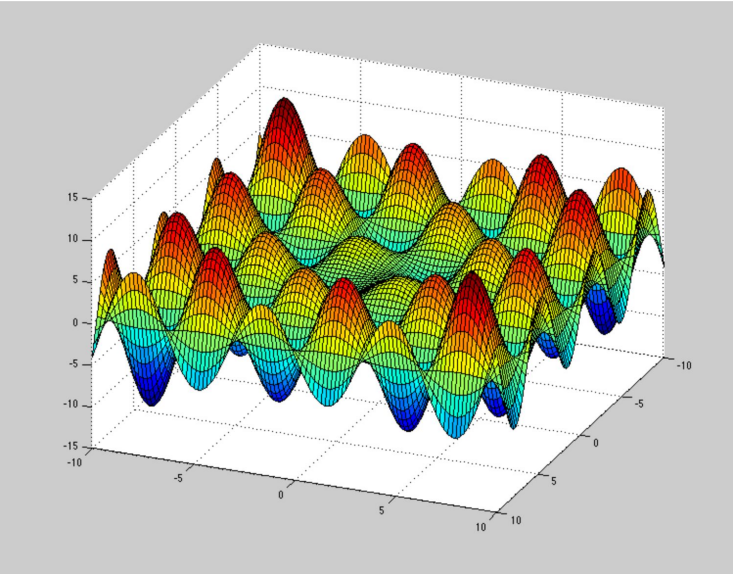
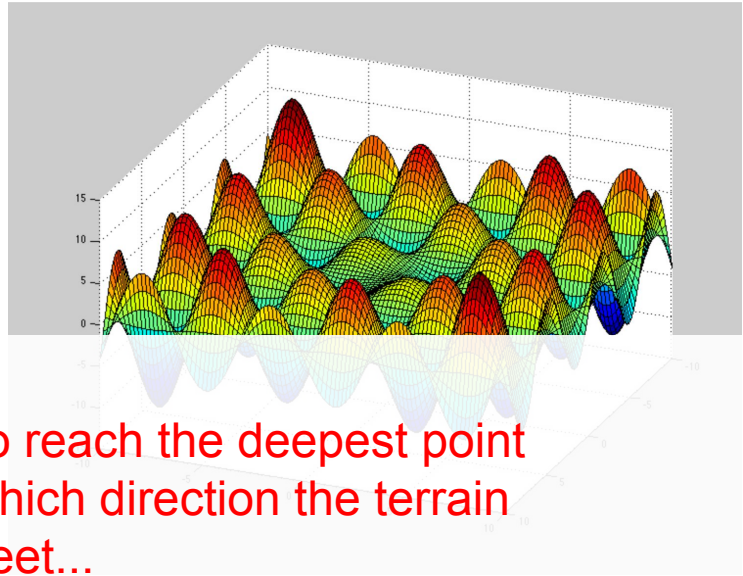


Image sources: spektr @ scicomp.stackexchange.com

Linear regression - The least squares method

Is it guaranteed that we will find the optimal solution
(minimum MSE loss) **for any function using gradient descent?**



No. We can reach
one of the **local minimum points**,
but if the loss function is **not convex**,
then it is **not guaranteed**
that this will be the **global minimum**.

Mountain hiking in fog: We want to reach the deepest point of the terrain, but we can only feel which direction the terrain slopes downwards most under our feet...