neue fische

School and Pool for Digital Talent

Gradient Descent

Motivation

Intro Gradient descent

Learning rate

Summary

neue fische
School and Pool for Digital Talent

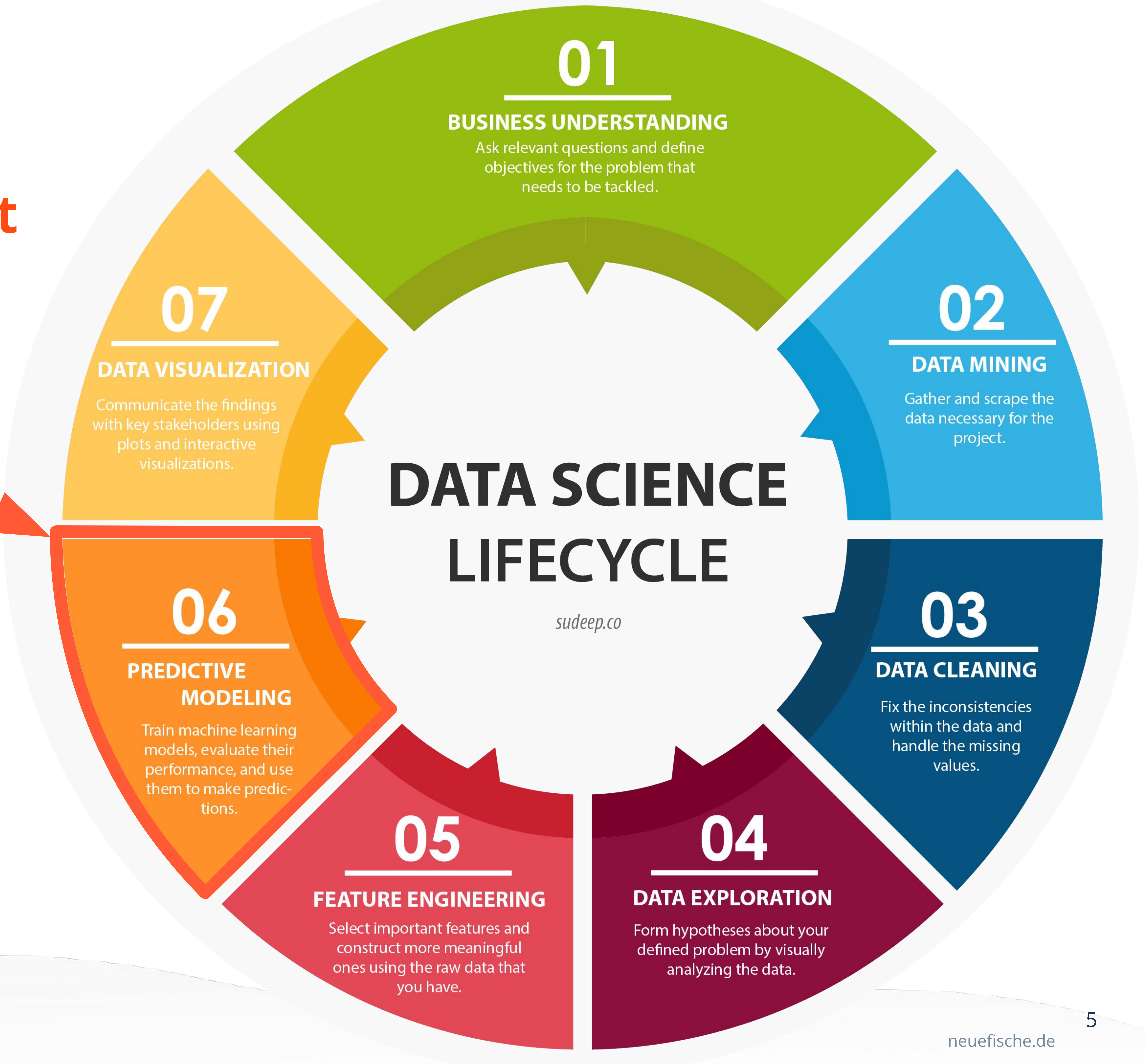
Gradient Descent

Part 1
Motivation

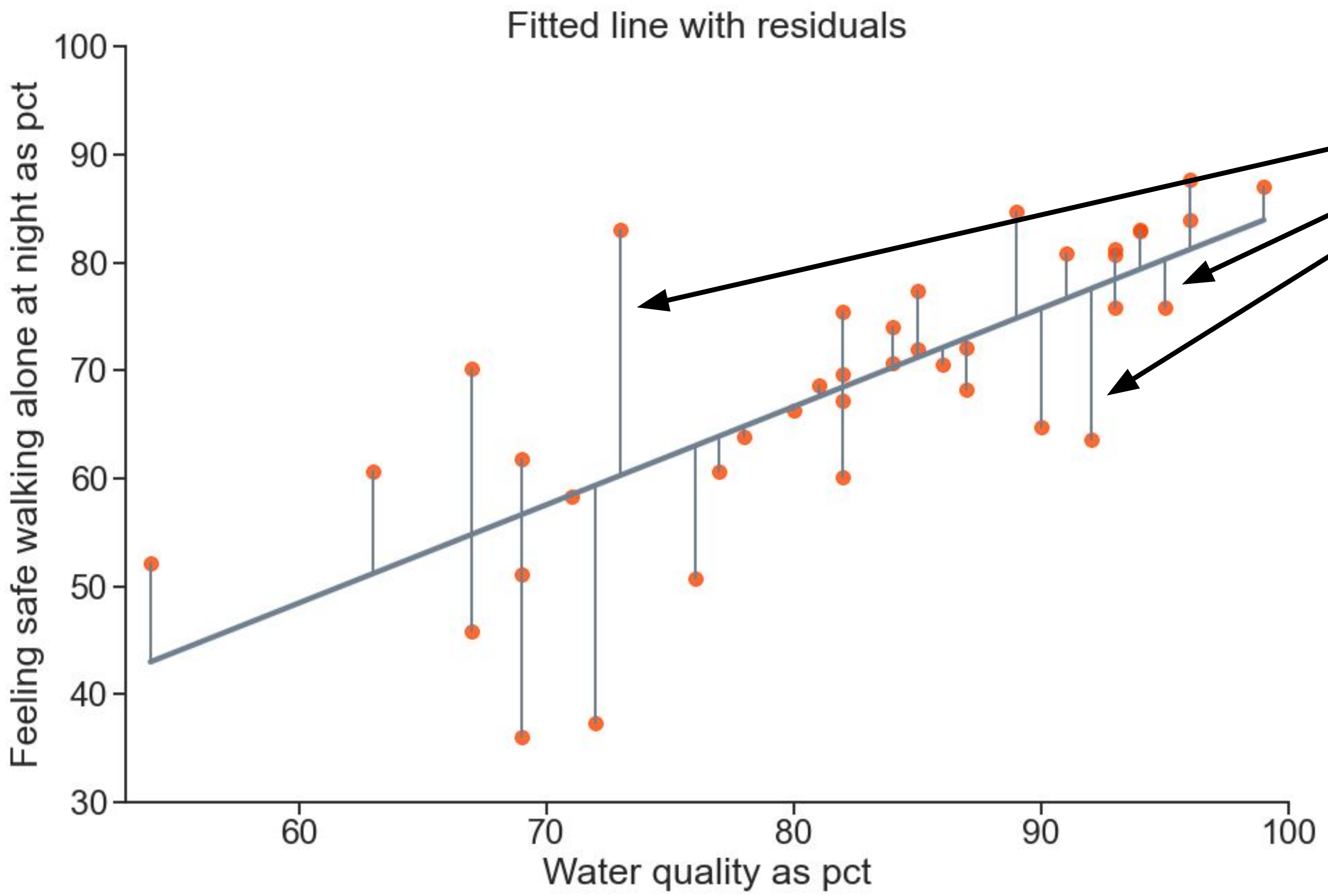
Where is Gradient Descent used in the Data Science Lifecycle?

06 PREDICTIVE MODELING:

- select a ML algorithm
- train the ML model
- evaluate the performance
- make predictions



Recap of Optimization of Linear Regression with OLS



Model $$y = b_0 + b_1 \cdot x + e$$

Residuals $$e_i = y_i - \hat{y}_i$$

OLS:

minimize the sum of squared residuals

$$\sum_{i=1}^{n} e_i^2$$

results in Normal Equation:

$$b = \left(X^T X\right)^{-1} X^T y$$

→ closed form solution

neue fische
School and Pool for Digital Talent

Gradient Descent

Part 2
Intro Gradient descent

Overview

Definition

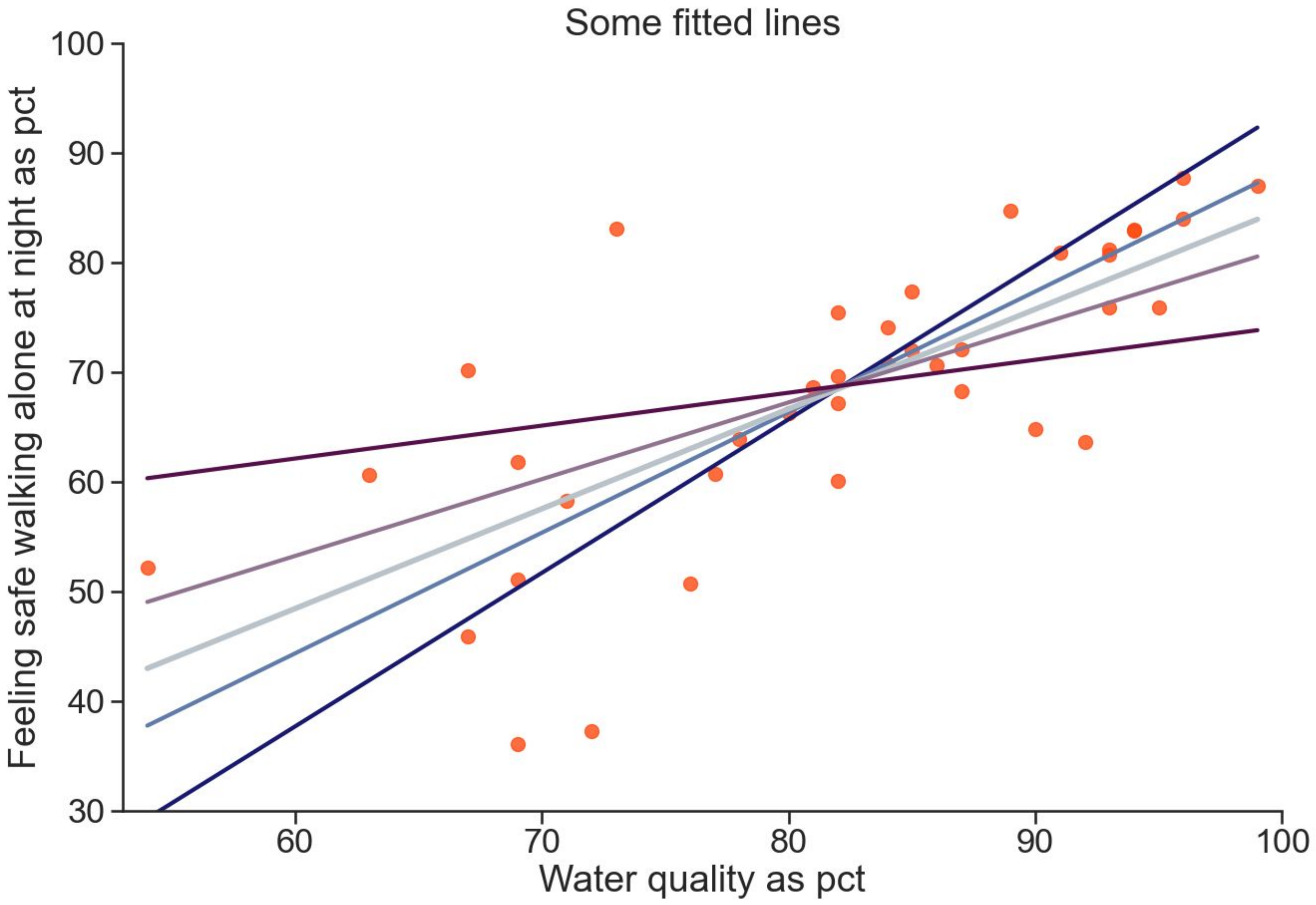
Gradient Descent is an **iterative optimization algorithm** to find the minimum of a function.

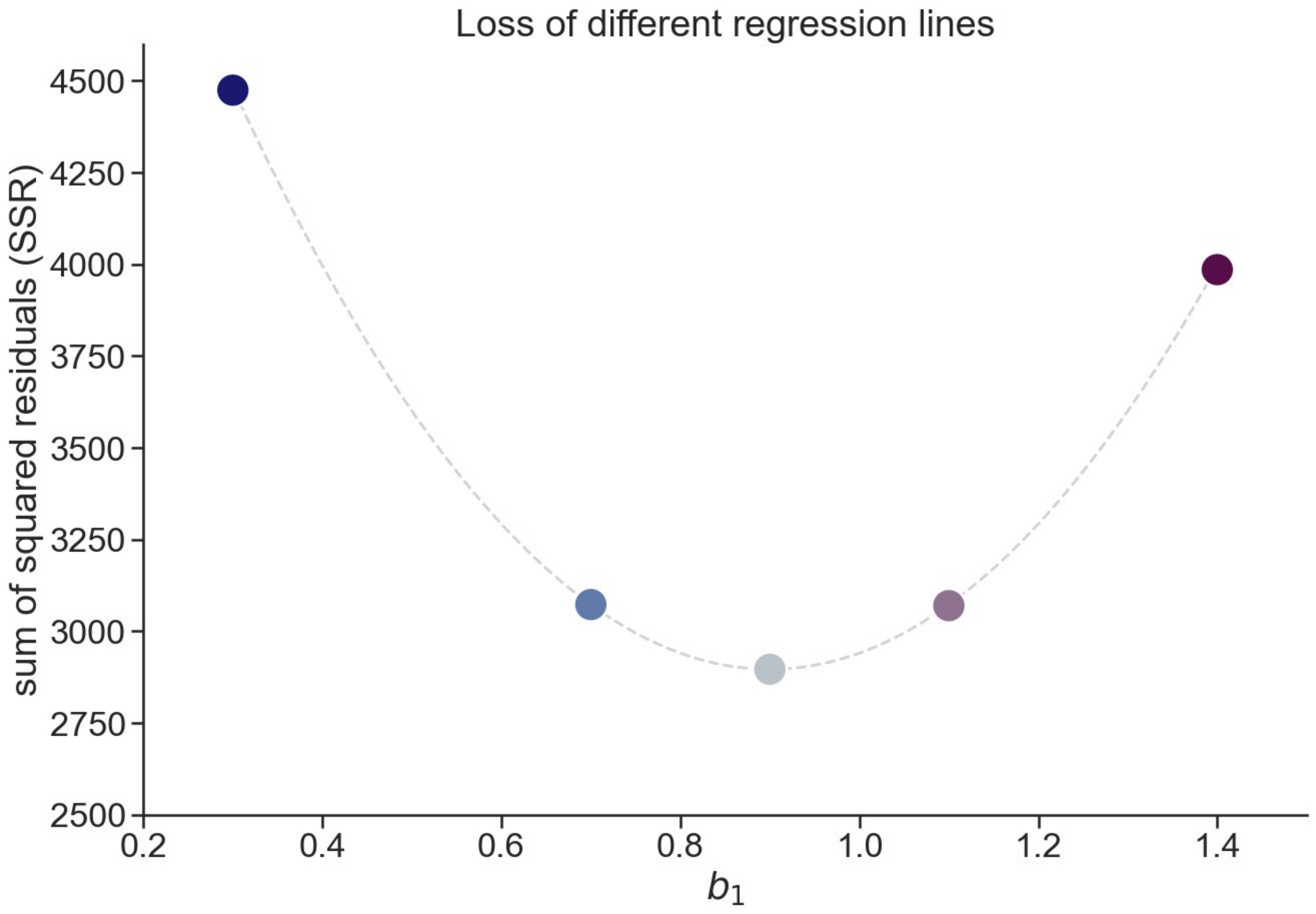
→ e.g. of a cost function

Training = Finding the best fitting line for our data

We get to the best fitting line by minimizing the cost function:

$$J(b_0, b_1) = \sum e_i^2 = \sum (y_i - \hat{y}_i)^2 = \sum (y_i - b_0 - b_1 x_i)^2$$





When to use Gradient Descent?

- GD is a simple **optimization procedure** that can be used for many machine learning algorithms (eg. linear regression, logistic regression, SVM)
- used for “**Backpropagation**” in Neural Nets (variations of GD)
- can be used for **online learning** (needed when Data cannot be stored on only one computer)
- gives **faster results** for problems with many features (computationally simpler) (double nr. of features: Normal Equation will take 5-8 times longer)

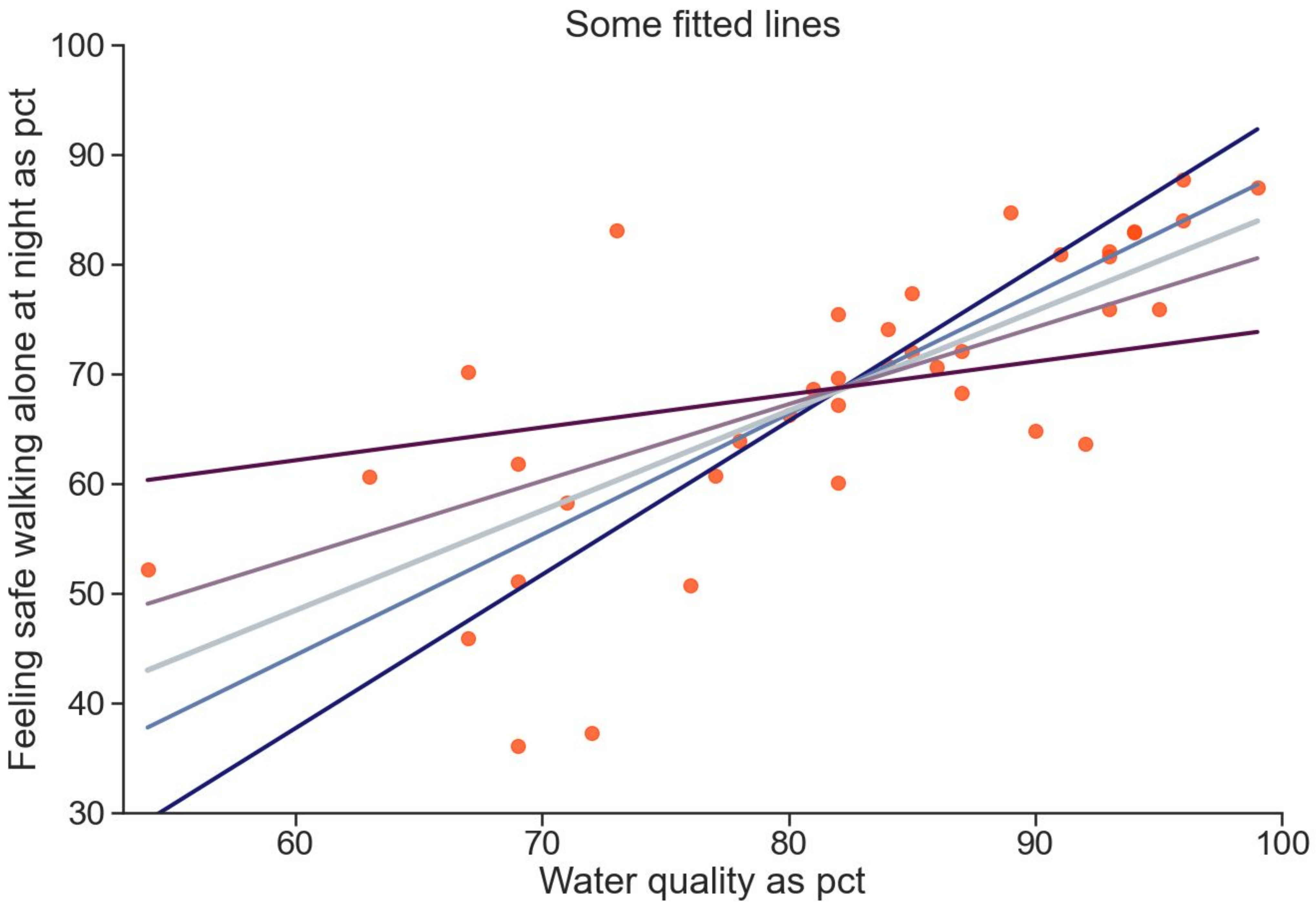
How to use Gradient Descent?

First Step

Define Model

Example:
least squares with one feature

$$\hat{y} = b_0 + b_1 \cdot x$$



Second Step

Define Cost Function

We get to the best fitting line by minimizing the cost function:

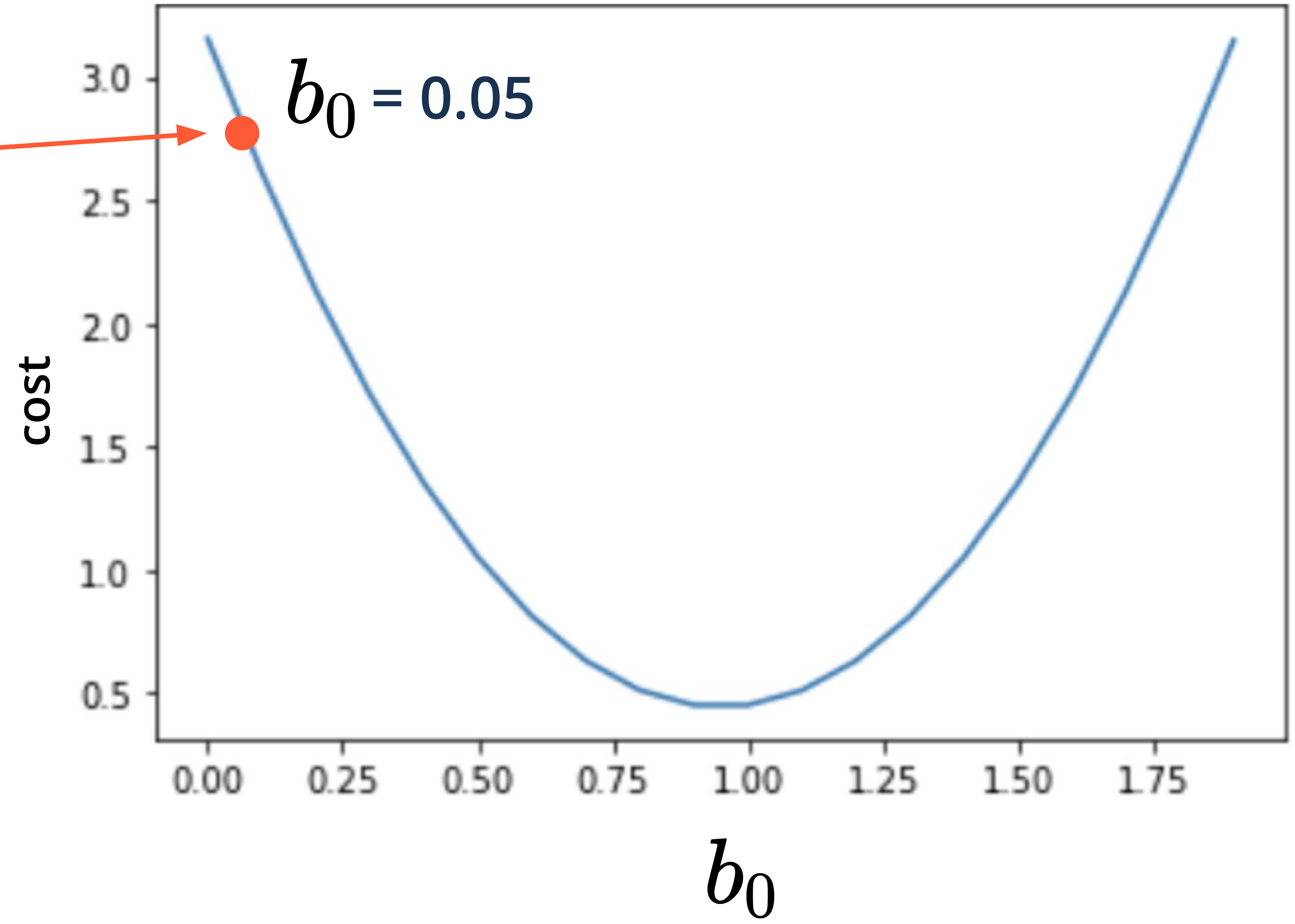
$$J(b_0, b_1) = \sum e_i^2 = \sum (y_i - \hat{y}_i)^2 = \sum (y_i - b_0 - b_1 x_i)^2$$

$$\min(J(b_0, b_1))$$

Third Step

Initialize Gradient Descent

Deliberately set some random starting values b



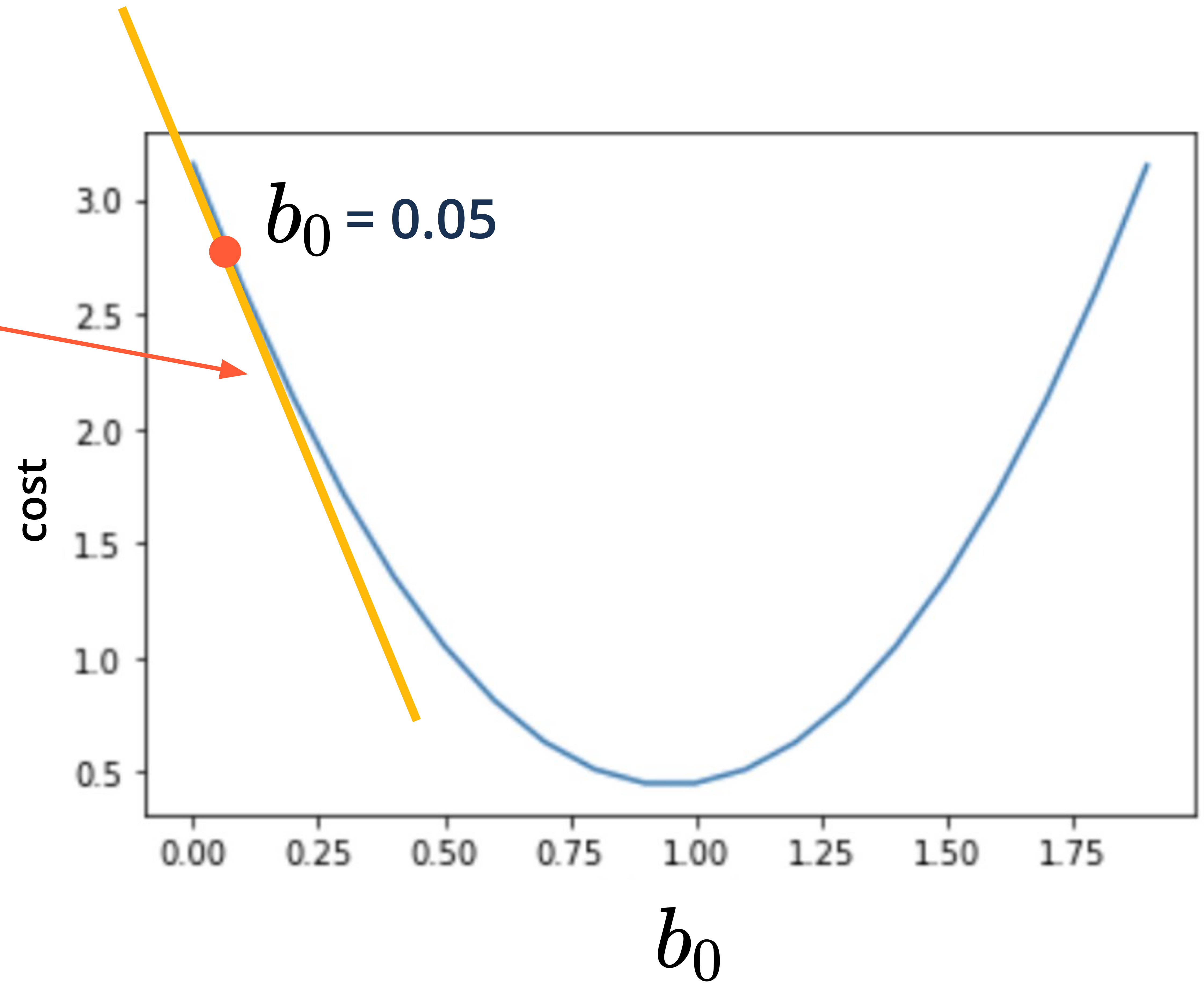
Fourth Step

Gradient Descent

Start descent:

- Take derivatives of the cost function with respect to your parameters b

$\boxed{slope = -5.7}$

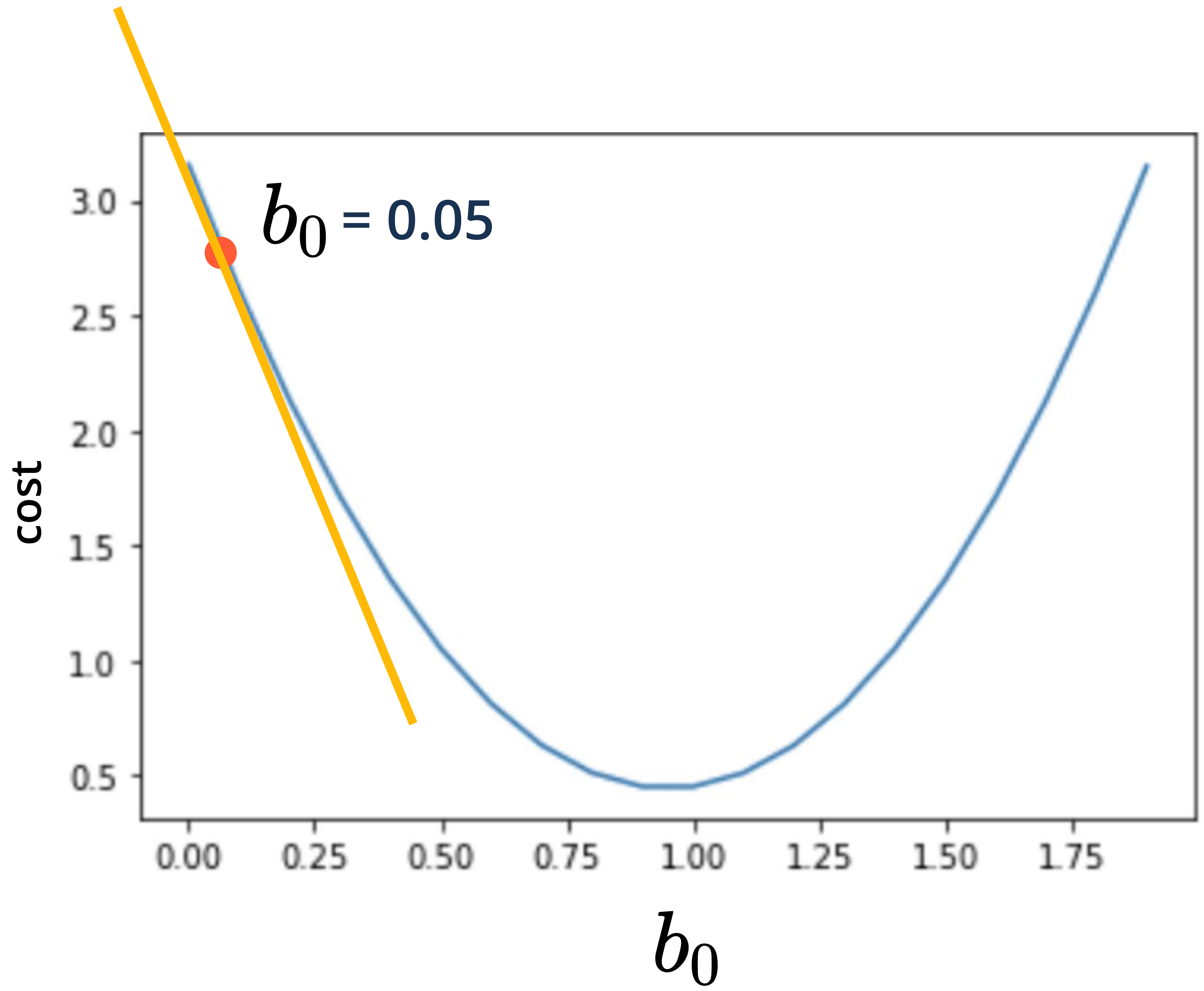


Gradient Descent

Start descent:

- Take derivatives with respect to your parameters b
- Set your learning rate (step-size)

$\boxed{\alpha = 0.1}$ $slope = -5.7$



Gradient Descent

Start descent:

- Take derivatives with respect to your parameters b
- Set your learning rate (step-size)
- Adjust your parameters (step)

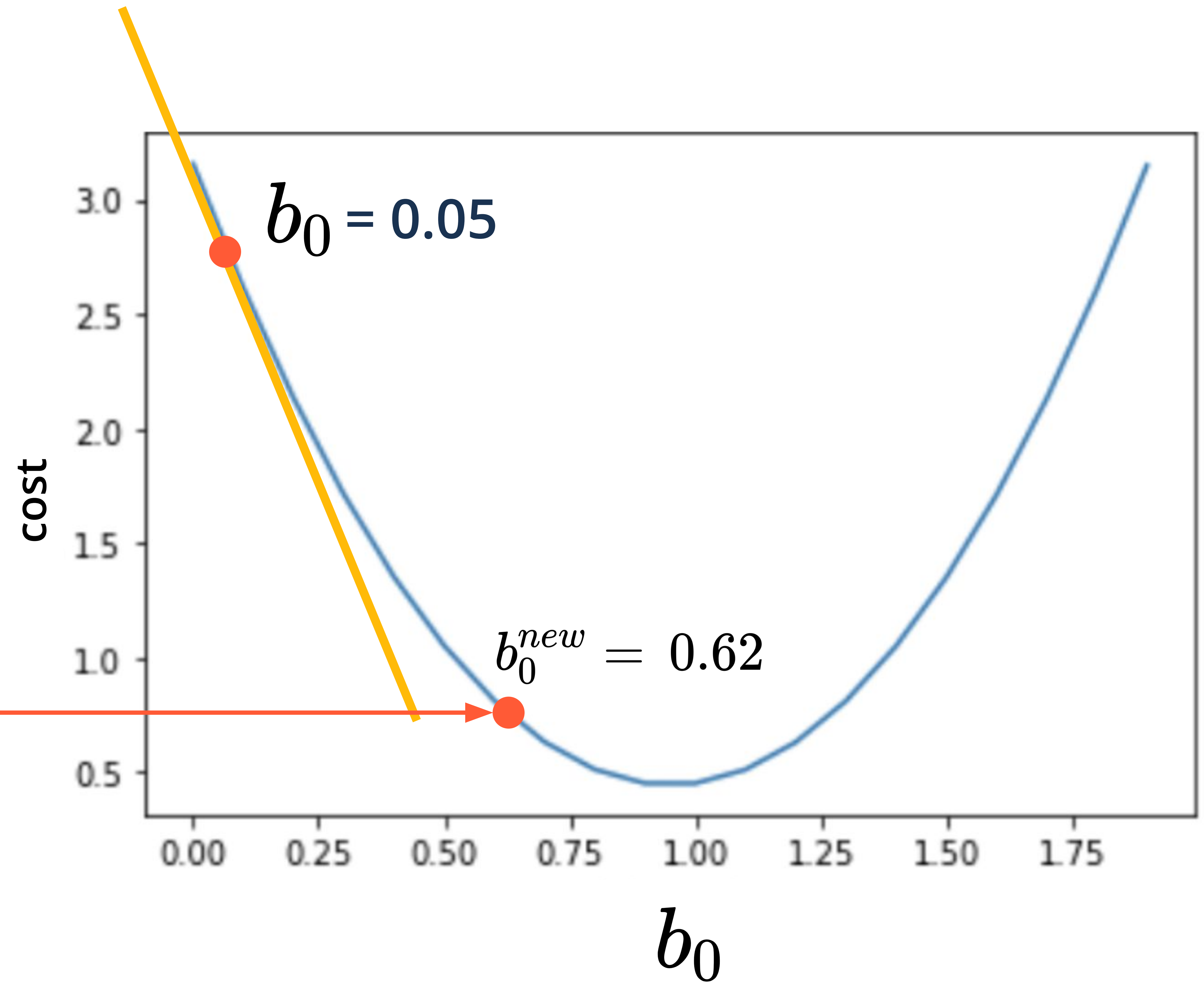
$$\alpha = 0.1 \qquad slope = -5.7$$

$$step = \alpha \cdot derivative\, of\, cost\, function$$

$$b_0^{new} = b_0^{old} - step$$

$$b_0^{new} = 0.05 - (0.1 \cdot (-5.7))$$

$$\boxed{b_0^{new} = 0.62}$$



Fifth Step

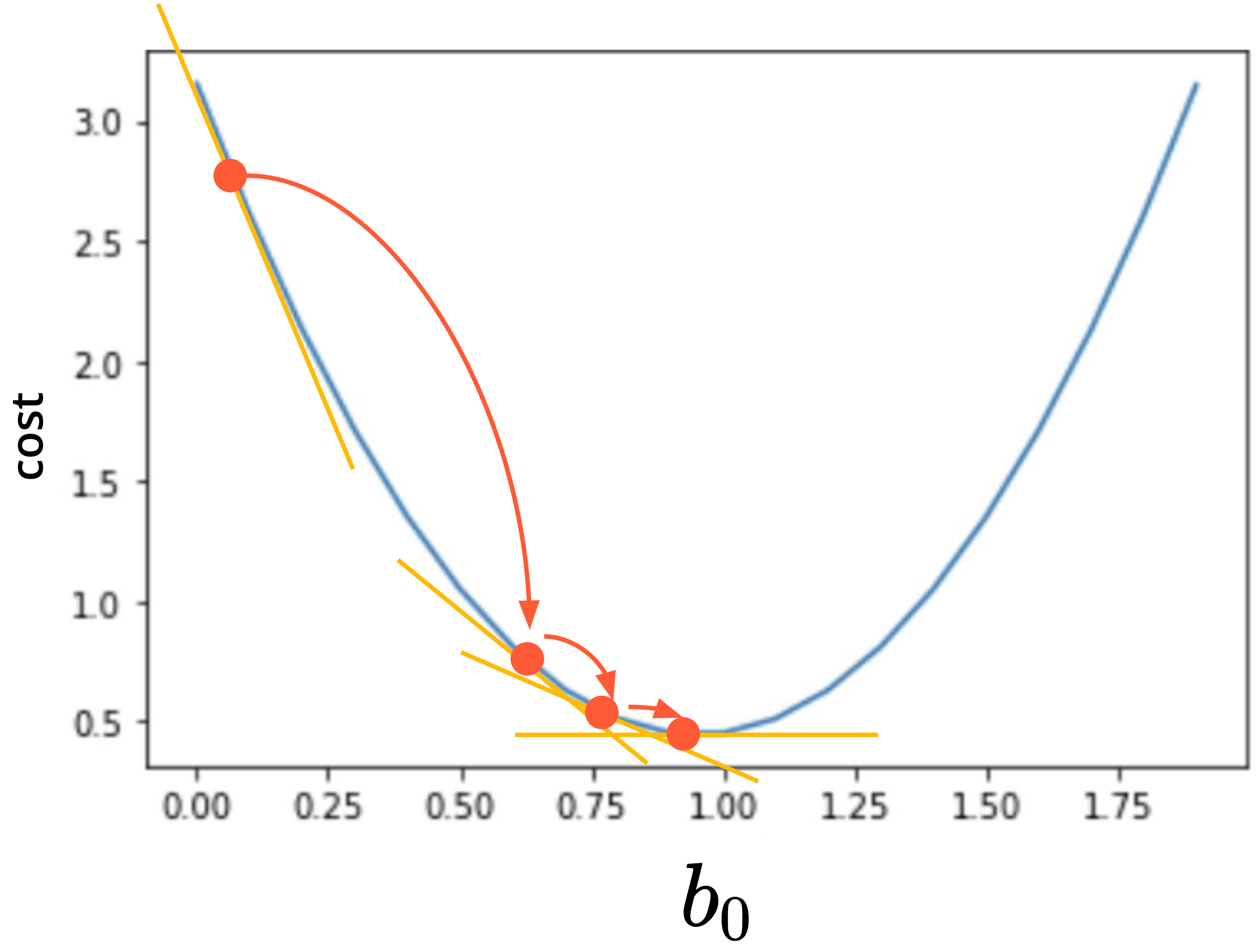
Gradient Descent

Start descent:

- Take derivatives with respect to your parameters b
- Set your learning rate (step-size)
- Adjust your parameters (step)

$$step = \alpha \cdot derivative\,of\,cost\,function$$
$$b_0^{new} = b_0^{old} - step$$

Repeat till there is no further improvement



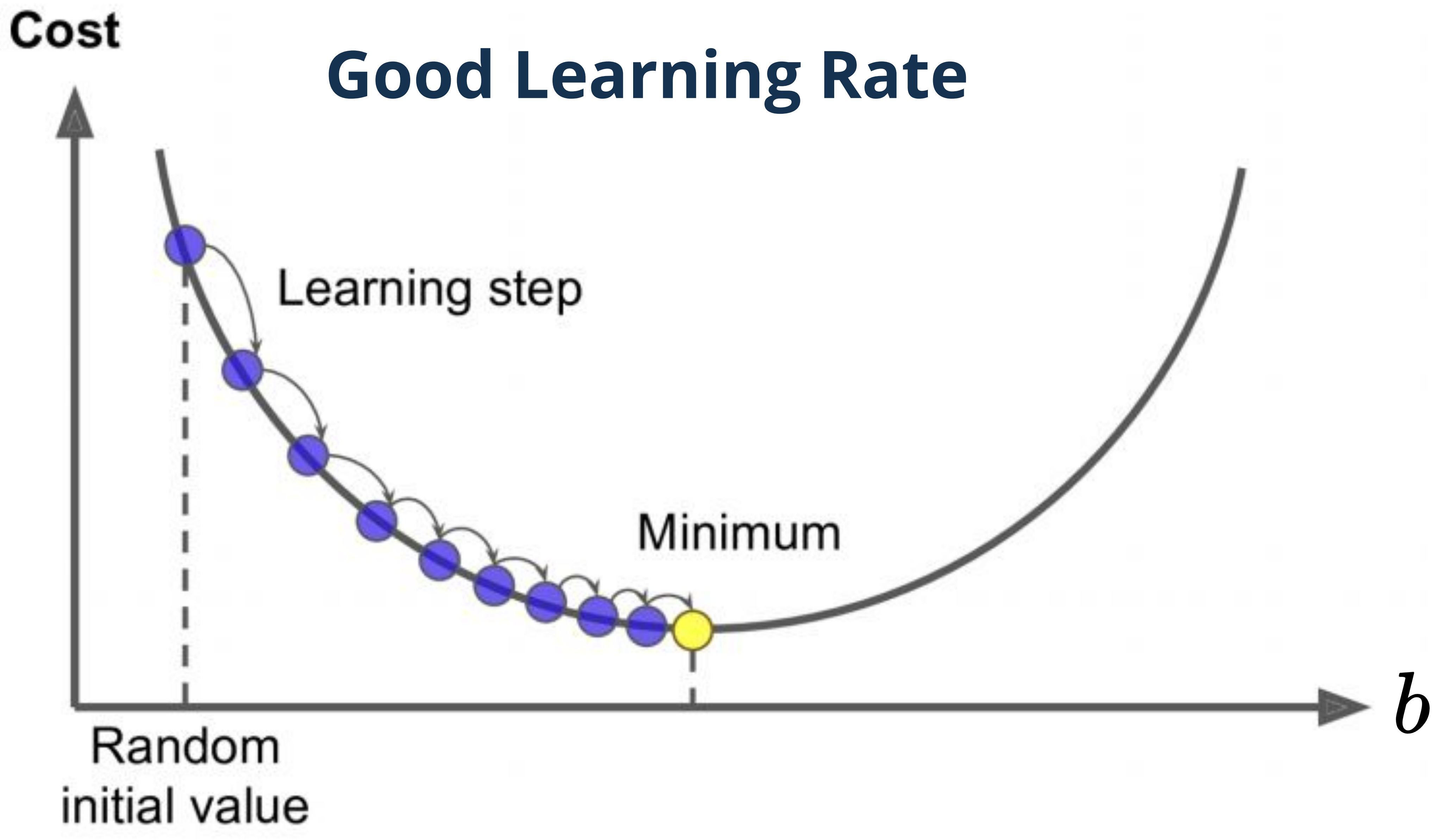
neue fische
School and Pool for Digital Talent

Gradient Descent

Part 3
Learning Rate

Learning Rate

It is important to find a good value for the learning rate!



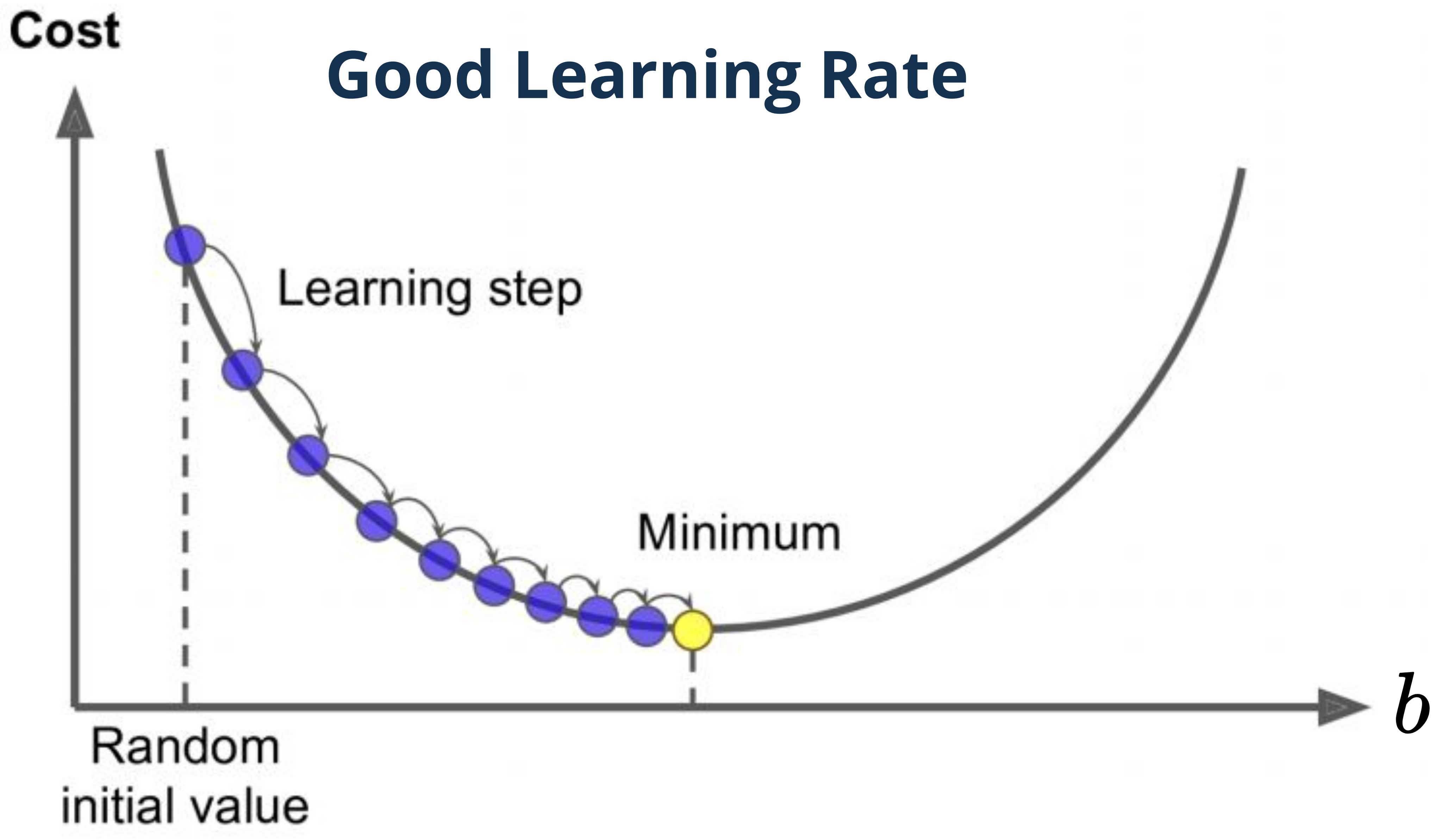
Source:
A. Geron, "Hands-on ML with scikit-learn and tensorFlow", 2017

Gradient Descent - Hyperparameter

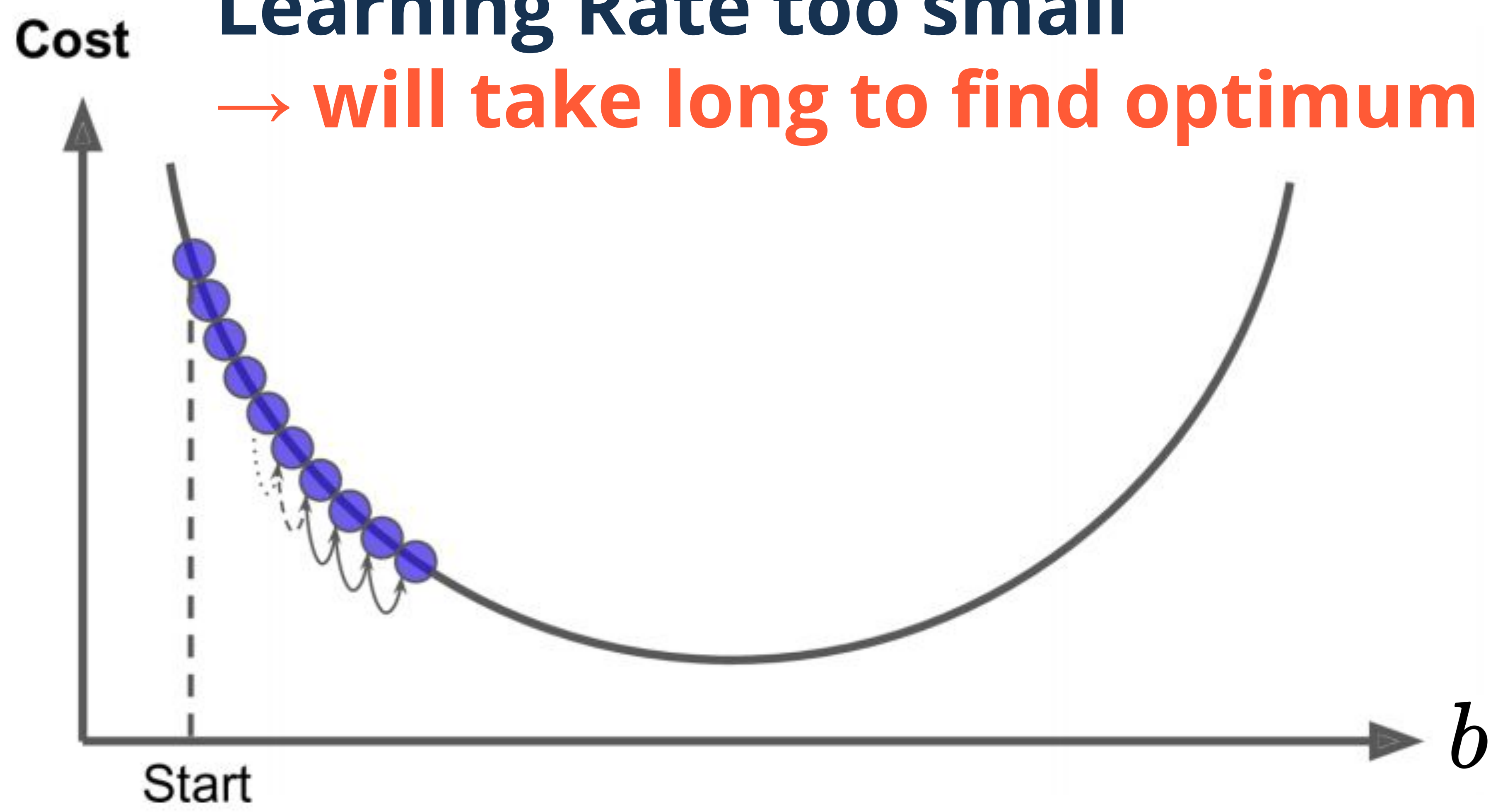
Learning Rate

It is important to find a good value for the learning rate!

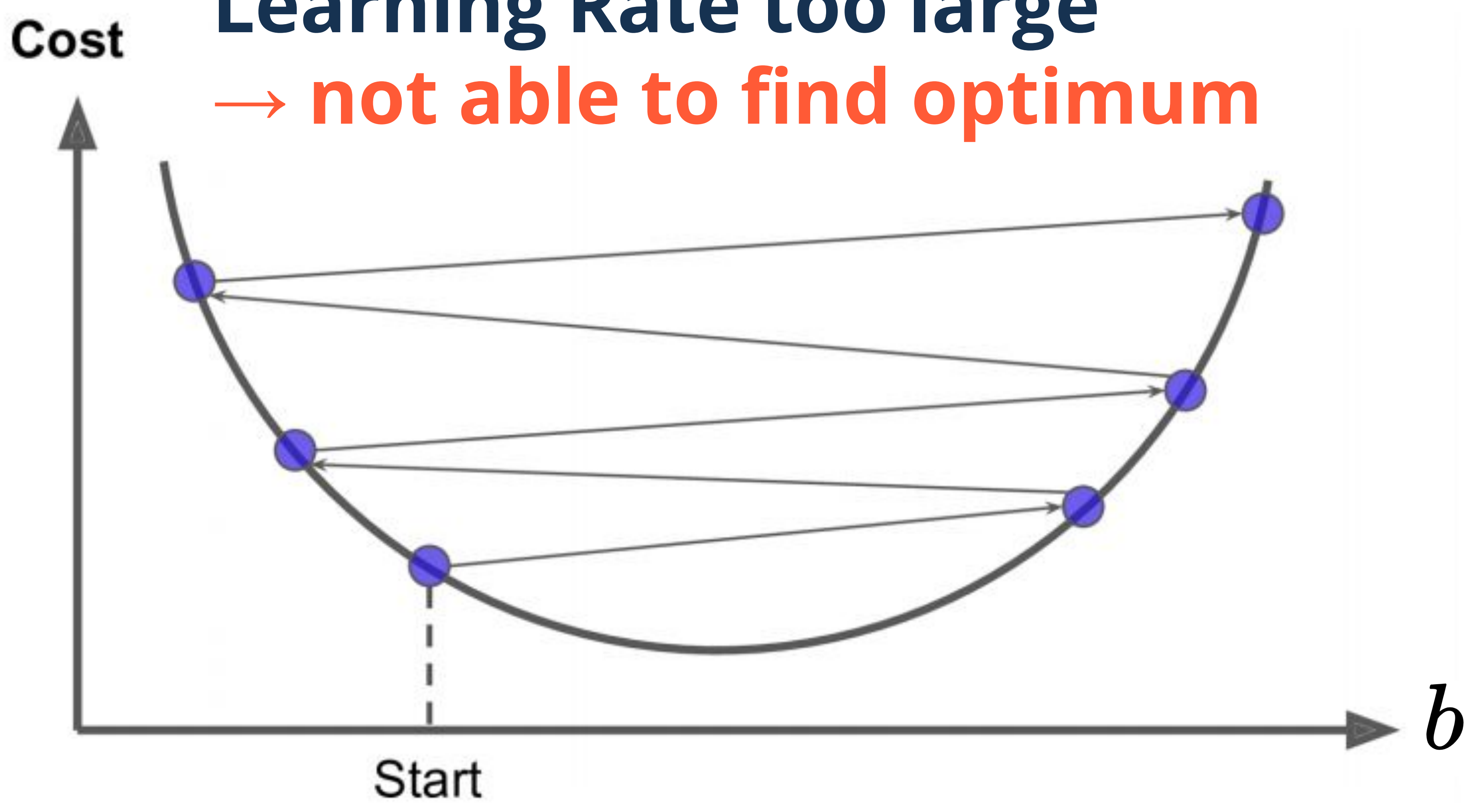
Good Learning Rate



Learning Rate too small
→ will take long to find optimum



Learning Rate too large
→ not able to find optimum



Source:
A. Geron, "Hands-on ML with scikit-learn and tensorFlow", 2017

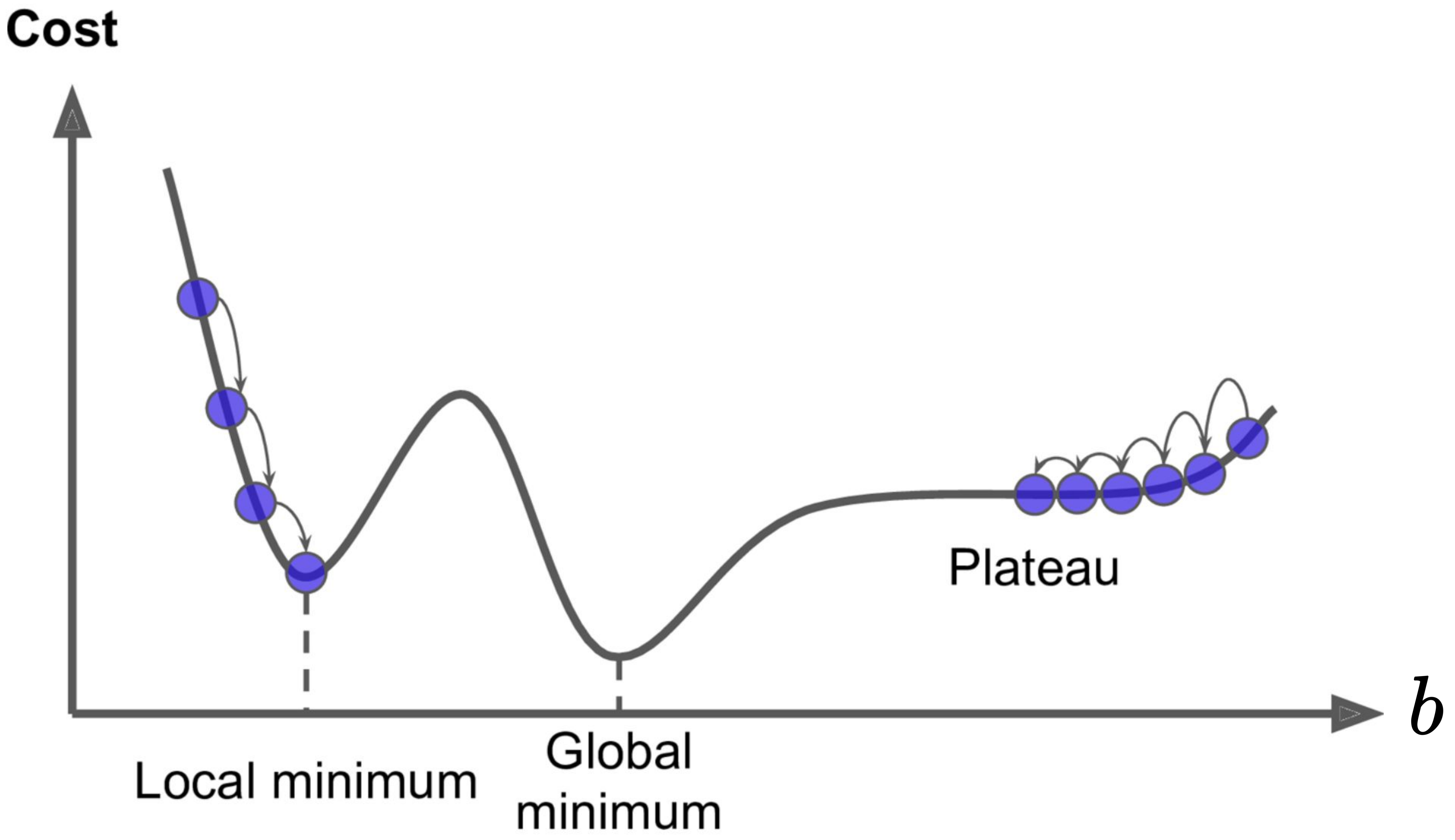
Problems

Two main challenges of Gradient Descent

The MSE cost function for linear regression is a convex function, it just has one global minimum.

This is not always the case. Problems are:

- local minima
- plateaus



Convex: If you pick two random points the line that is connecting them will never below the curve.

Source:
A. Geron, "Hands-on ML with scikit-learn and tensorFlow", 2017

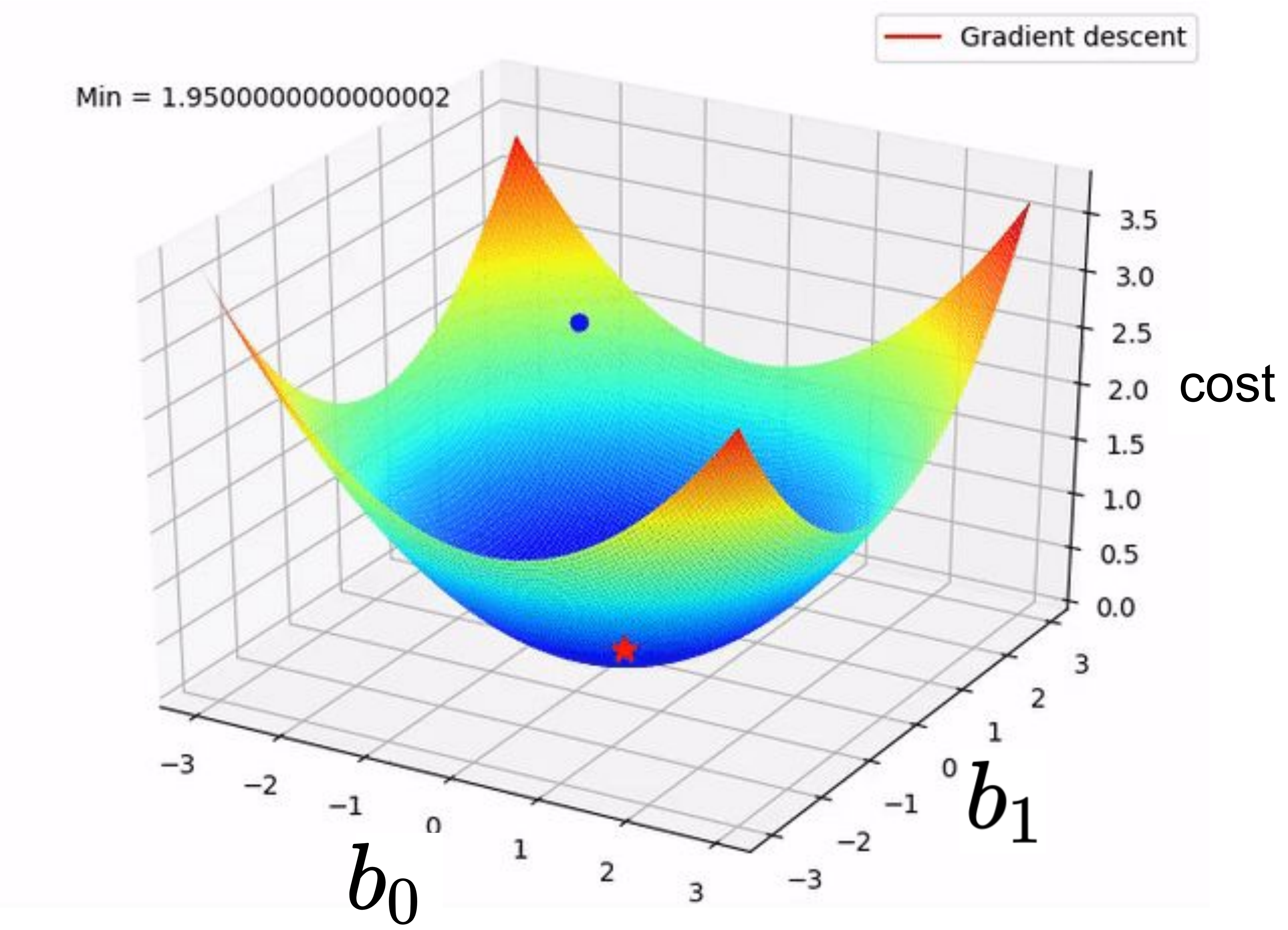
neue fische
School and Pool for Digital Talent

Gradient Descent

Part 4
Summary

Gradient Descent summed up

1. Define model
2. Define the cost function
3. Deliberately set some starting values
4. Start descent:
   - Take derivatives with respect to parameters
   - Set your learning rate (step-size)
   - Adjust your parameters (step)
5. Repeat 4. till there is no further improvement



Source: https://suniljangirblog.wordpress.com/2018/12/03/the-outline-of-gradient-descent/

References

[Gradient Descent Step by Step]