Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media

Regression

With Linear Regression

MODALG SoSe18

News & Motivation

META MARATHON
ARTIFICIAL INTELLIGENC

42 radical-explorative hours
NRW-Forum Düsseldorf
May 25 — 27, 2018

Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media

42 Stunden nonstop Talks, Performances, Filme, Konzerte, Ausstellung und Workshops zum **Thema Künstliche Intelligenz**: Der META Marathon ist ein neuartiges Technologie-Festival, das vom 25. bis 27. Mai 2018 im NRW-Forum Düsseldorf stattfindet. Die Teilnehmer gestalten das Festival selbst, wechseln die Rollen vom Experten zum Laien und experimentieren mit dem neuen Format – **inklusive Übernachtung vor Ort**. Festivaldirektor ist der Futurist und Unternehmer Christopher Peterka.

Advantages:

- Participants who can show a **three day participation** receive **three points** for this class
- Three days for free – **do it!!!**
- **Networking** and learn cool **AI-Stuff**
- Our Workshop: Reinforcement-Learning with DQN‘s
- Timeslot: Sa. 26.5., 11 am. to 15 pm.

Important:

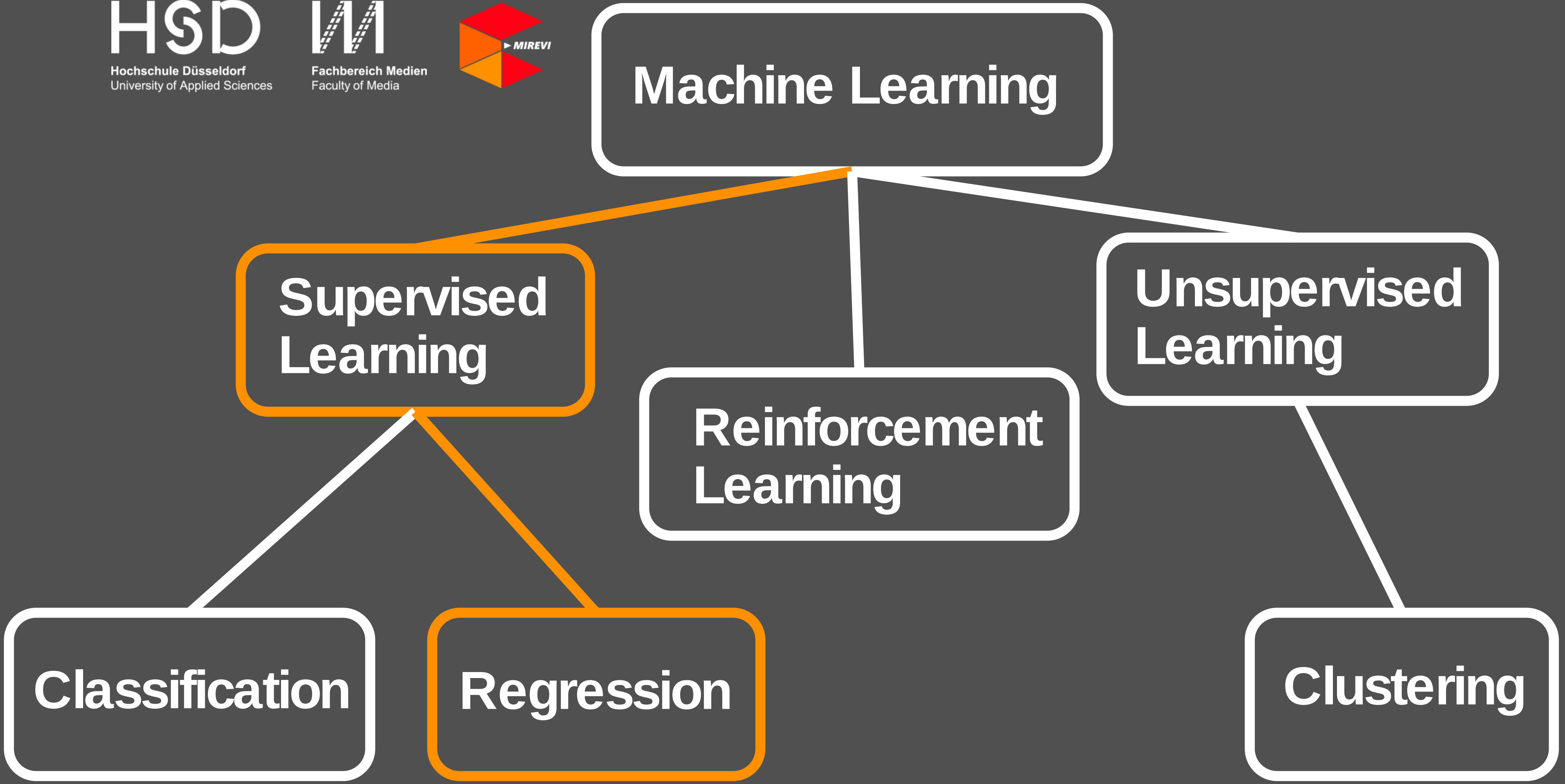
- Send me your confirmation and your name till 16.05.18, 18 pm.
- marcel.tiator@study.hs-duesseldorf.de
- Use subject: „META Marathon“
- More Information: https://www.nrw-forum.de/veranstaltungen/meta-marathon


name[get]script
response?
(245,23,068,789,a48)
if
#key_input
// script src= address
status.command
input. [true]
if ("true") add.string<status>
input.false function
response?
if
#key_input
if
script src= [true]

NEWCASTLE UNITED

Topics:

- Supervised Learning & Classification
- Regression
- Linear Regression
- Gradient Descent
- Introduction Tensorflow
- Programming Exercise


Machine Learning
Supervised Learning
Reinforcement Learning
Unsupervised Learning
Classification
Regression
Clustering

$$y = f(x)$$

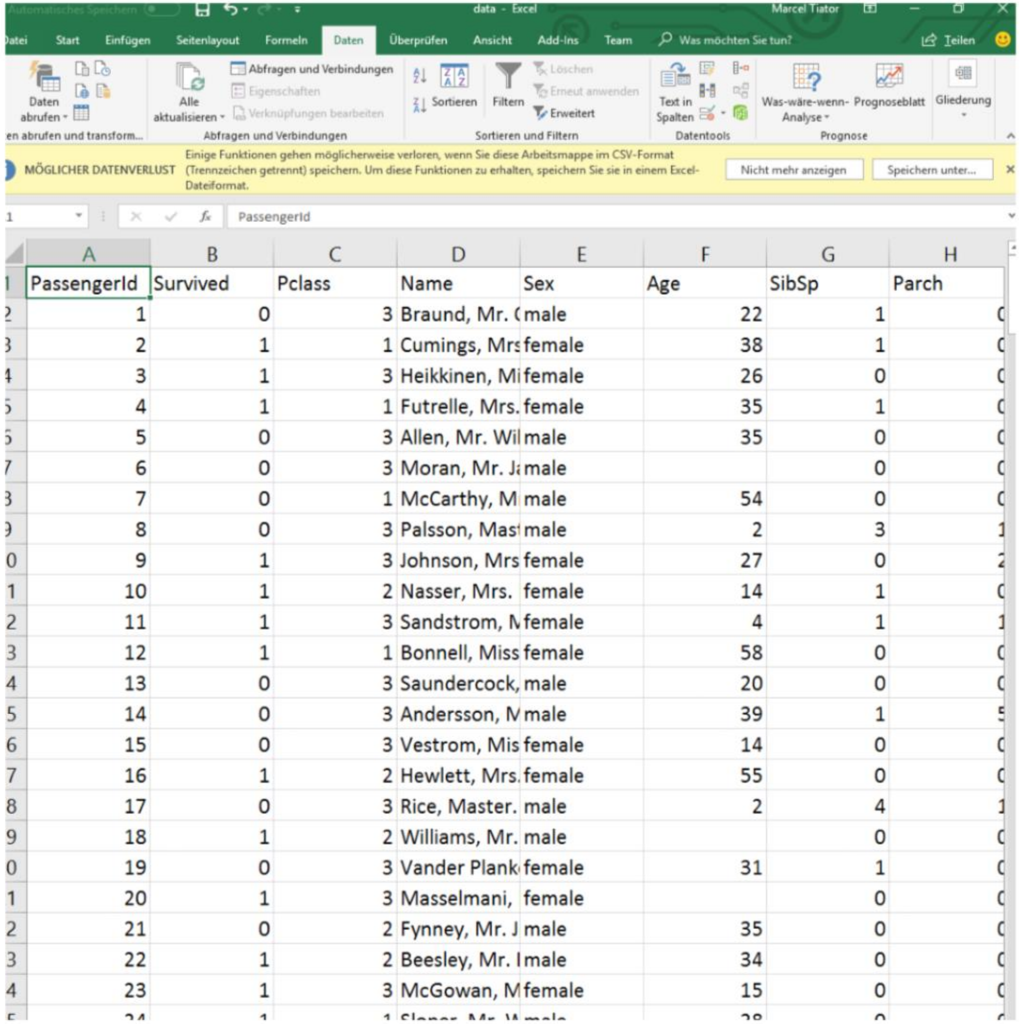
What is the aim of classification?

- **Differentiation**
- $y \in \{Partymaus, Spießer, Freak\}$

Tabular Data

$y: Survived$

$y \in \{0,1\}$

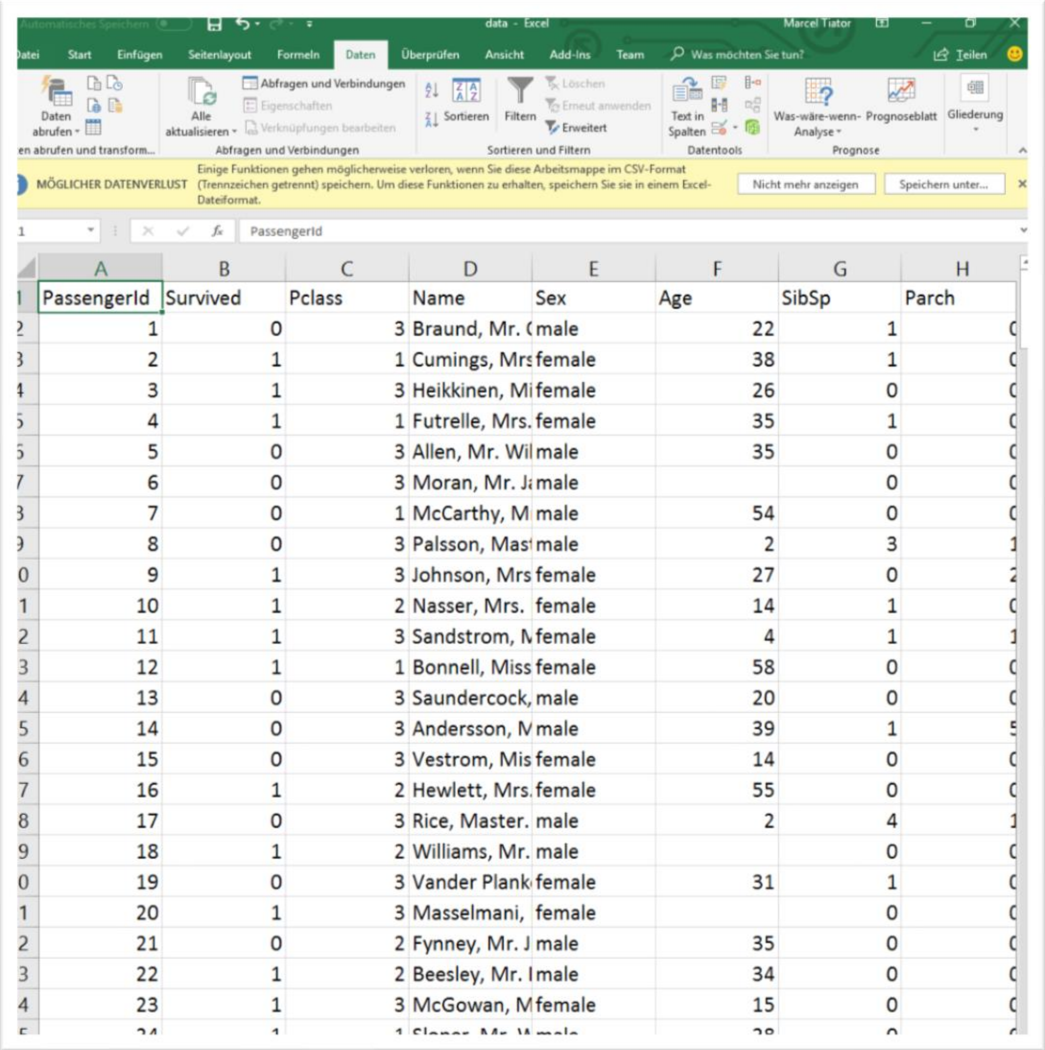


PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. (	male	22	1	(
2	1	1	Cumings, Mrs	female	38	1	(
3	1	3	Heikkinen, Mi	female	26	0	(
4	1	1	Futrelle, Mrs.	female	35	1	(
5	0	3	Allen, Mr. Wil	male	35	0	(
6	0	3	Moran, Mr. Ja	male		0	(
7	0	1	McCarthy, M	male	54	0	(
8	0	3	Palsson, Mast	male	2	3	1
9	1	3	Johnson, Mrs	female	27	0	2
10	1	2	Nasser, Mrs.	female	14	1	(
11	1	3	Sandstrom, M	female	4	1	1
12	1	1	Bonnell, Miss	female	58	0	(
13	0	3	Saundercock,	male	20	0	(
14	0	3	Andersson, M	male	39	1	5
15	0	3	Vestrom, Mis	female	14	0	(
16	1	2	Hewlett, Mrs.	female	55	0	(
17	0	3	Rice, Master.	male	2	4	1
18	1	2	Williams, Mr.	male		0	(
19	0	3	Vander Plank	female	31	1	(
20	1	3	Masselmani,	female		0	(
21	0	2	Fynney, Mr. J	male	35	0	(
22	1	2	Beesley, Mr. I	male	34	0	(
23	1	3	McGowan, M	female	15	0	(

Tabular Data

$x: class, sex, age$

e.g. $x = (1,1,45)^T$



PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
1	0	3	Braund, Mr. (	male	22	1	0
2	1	1	Cumings, Mrs	female	38	1	0
3	1	3	Heikkinen, Mi	female	26	0	0
4	1	1	Futrelle, Mrs.	female	35	1	0
5	0	3	Allen, Mr. Wil	male	35	0	0
6	0	3	Moran, Mr. Ja	male		0	0
7	0	1	McCarthy, M	male	54	0	0
8	0	3	Palsson, Mast	male	2	3	1
9	1	3	Johnson, Mrs	female	27	0	2
10	1	2	Nasser, Mrs.	female	14	1	0
11	1	3	Sandstrom, M	female	4	1	1
12	1	1	Bonnell, Miss	female	58	0	0
13	0	3	Saundercock,	male	20	0	0
14	0	3	Andersson, M	male	39	1	5
15	0	3	Vestrom, Mis	female	14	0	0
16	1	2	Hewlett, Mrs.	female	55	0	0
17	0	3	Rice, Master.	male	2	4	1
18	1	2	Williams, Mr.	male		0	0
19	0	3	Vander Plank	female	31	1	0
20	1	3	Masselmani,	female		0	0
21	0	2	Fynney, Mr. J	male	35	0	0
22	1	2	Beesley, Mr. I	male	34	0	0
23	1	3	McGowan, M	female	15	0	0

Discrete vs. Continuous

$$y \in N^n$$

Example:

$$y \in \{0,1,2,3,4\}$$

$$y \in \{(1,0,0),$$
$$(0,1,0),$$
$$(0,0,1)\}$$

$$y \in R^n$$

Example:

$$y \in \{0.1, -0.43, 3.44\}$$

$$y \in \{(-0.2, 0.3, -0.7),$$
$$(3.1, 5.0, -0.62),$$
$$(0.44, -0.4, 4.5)\}$$

Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media

Regression

y: DAX


13.100
13.050
13.000
12.950
12.900
12.850
12.800
12.750
12.700
12.650
12.600
+2%
+1,7%
+1,5%
+1%
+0,5%
0%
-0,5%
-1%
03.05.18
04.05.18
07.05.18
08.05.18
09.05.18
10.05.18
11.05.18

Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

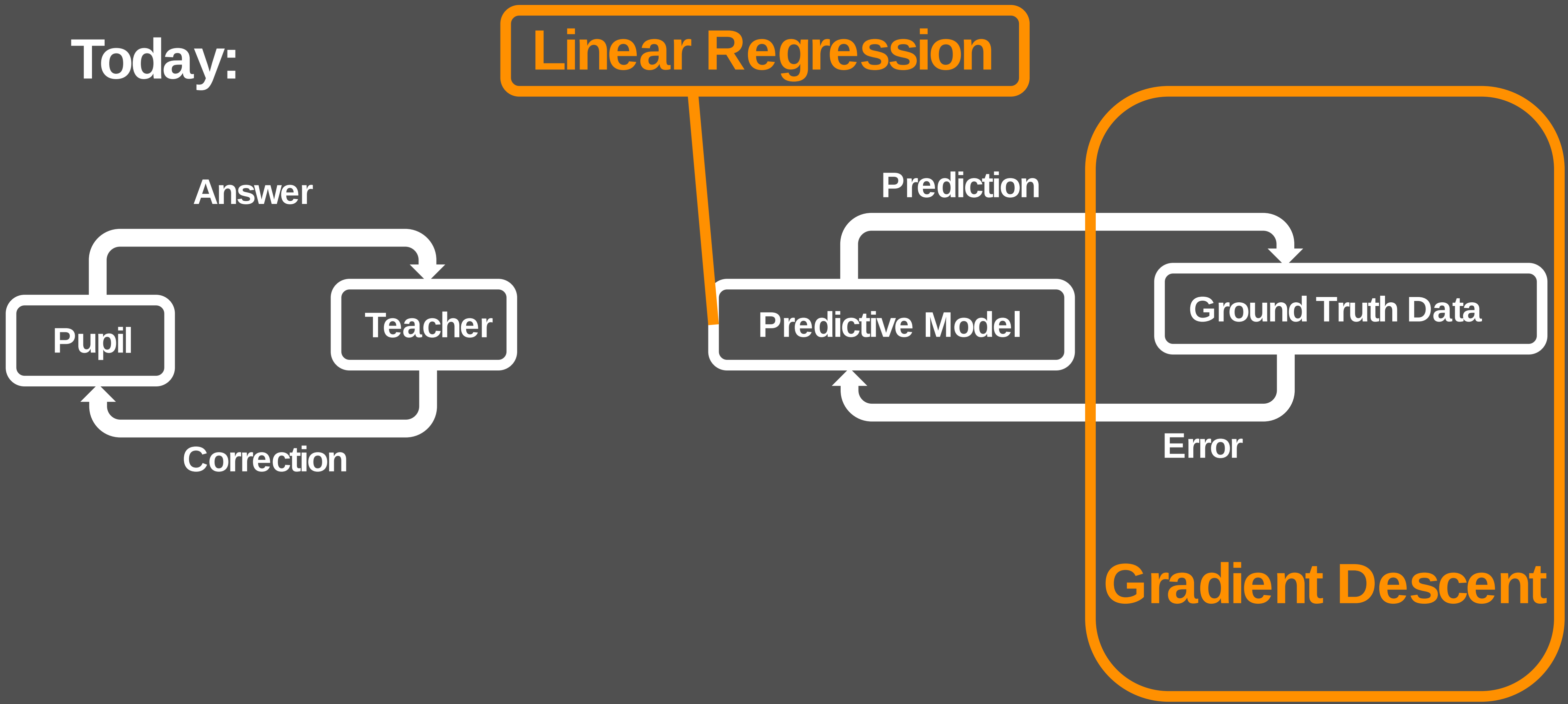
y: DAX

x: Dow Jones


25.000
24.900
24.800
24.700
24.600
24.500
24.400
24.300
24.200
+2%
+1,6%
+1%
0%
08.05.18
09.05.18
10.05.18
11.05.18

Date	x: Dow Jones	y: DAX
08.05.18	24,300	12,940
09.05.18	24,400	12,910
10.05.18	24,600	12,975
11.05.18	24,800	13,020
12.05.18	24,600	???

Today:


Linear Regression
Answer
Pupil
Teacher
Correction
Prediction
Predictive Model
Ground Truth Data
Error
Gradient Descent

Linear Regression

Predictive Model:

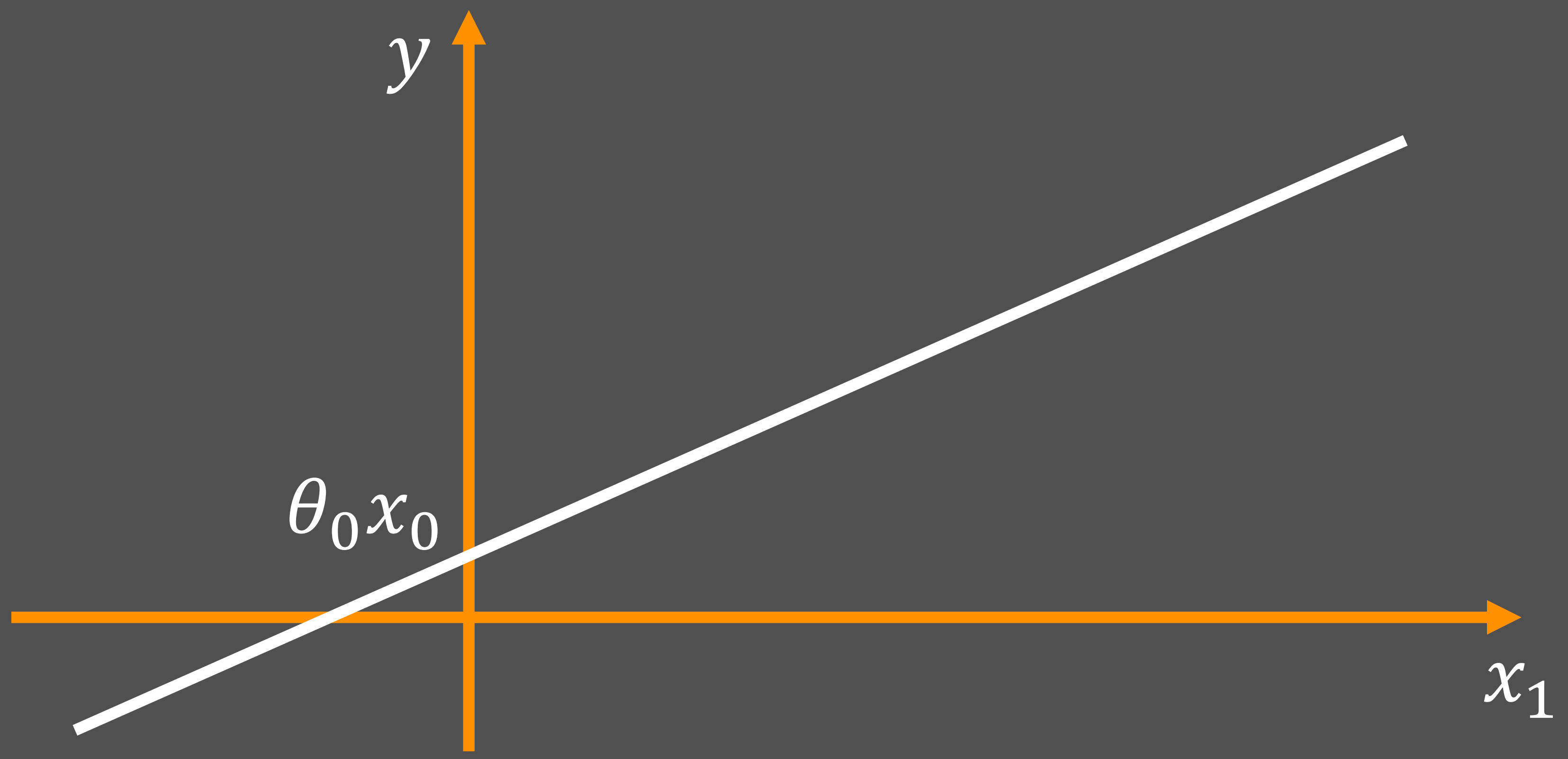
- $\hat{y} = \theta_1 x_1 + \theta_0 x_0$
- Weights: $\theta_0, \theta_1 \in R$
- $x_0 = 1$

Predictive Model:

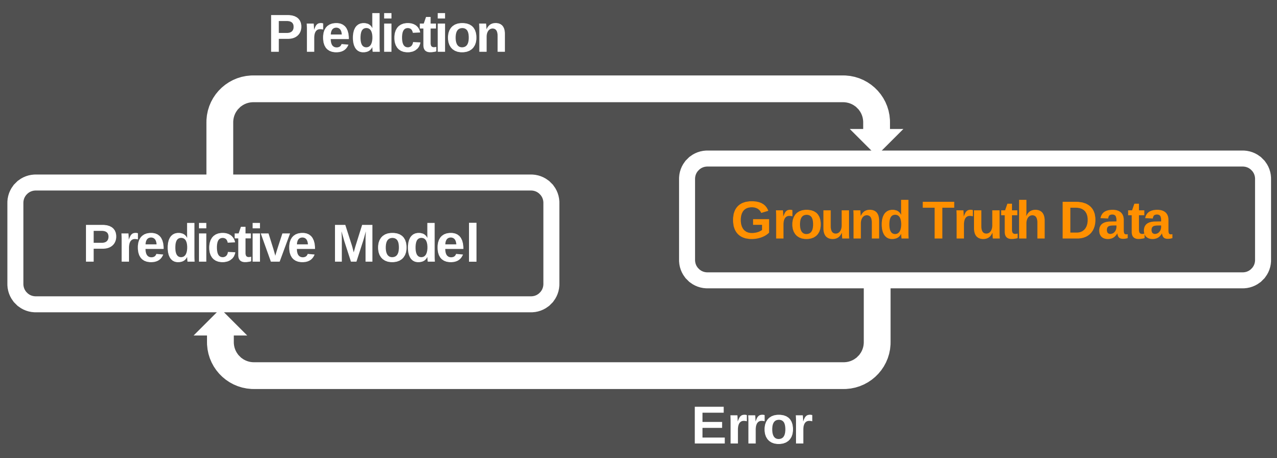
- $\hat{y} = mx + b$
- $m = \theta_1 x_1$
- $b = \theta_0 x_0$

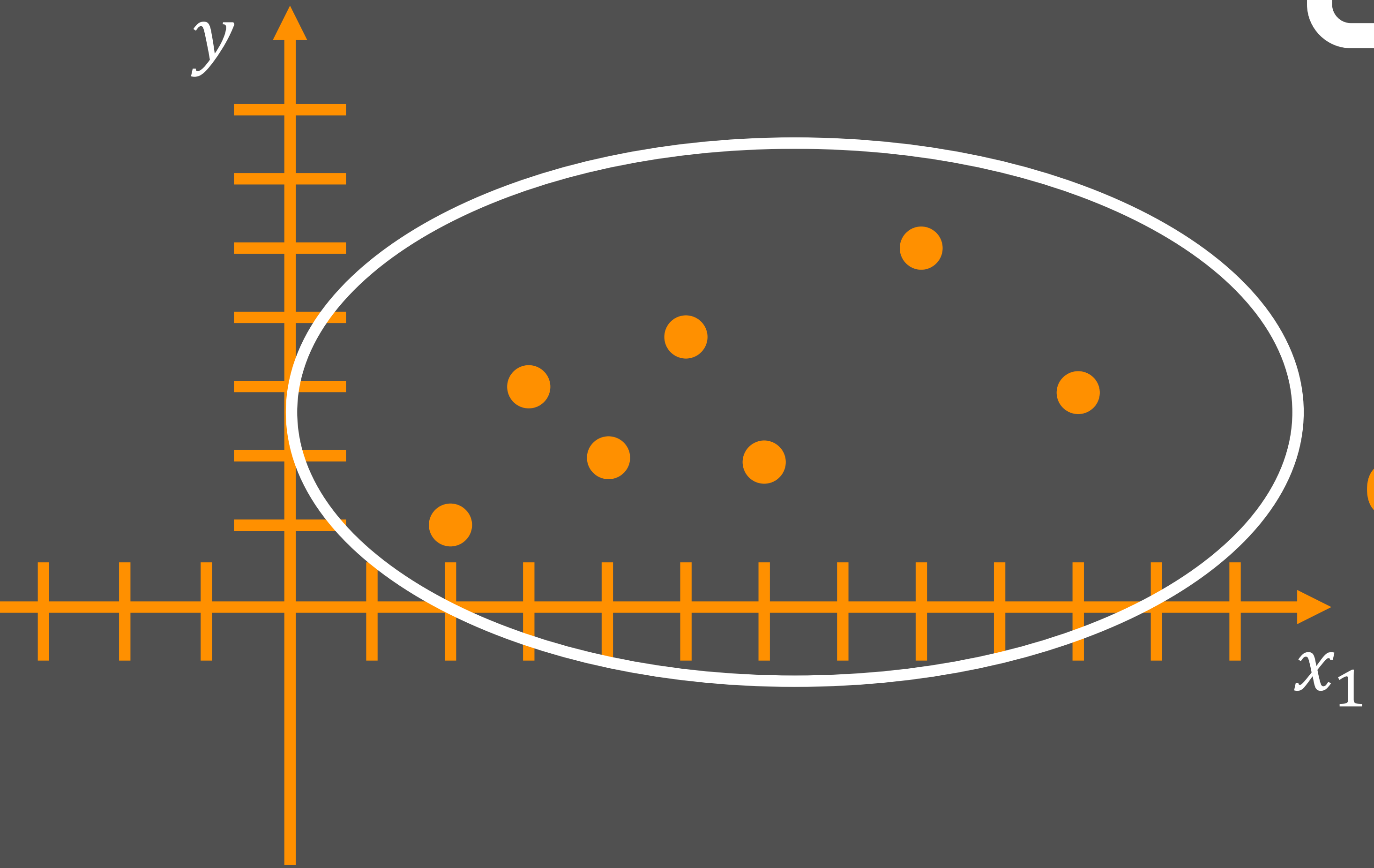
Slope of the line depends on weights θ_0, θ_1

$$\hat{y} = \theta_1 x_1 + \theta_0 x_0$$



Find good weights!

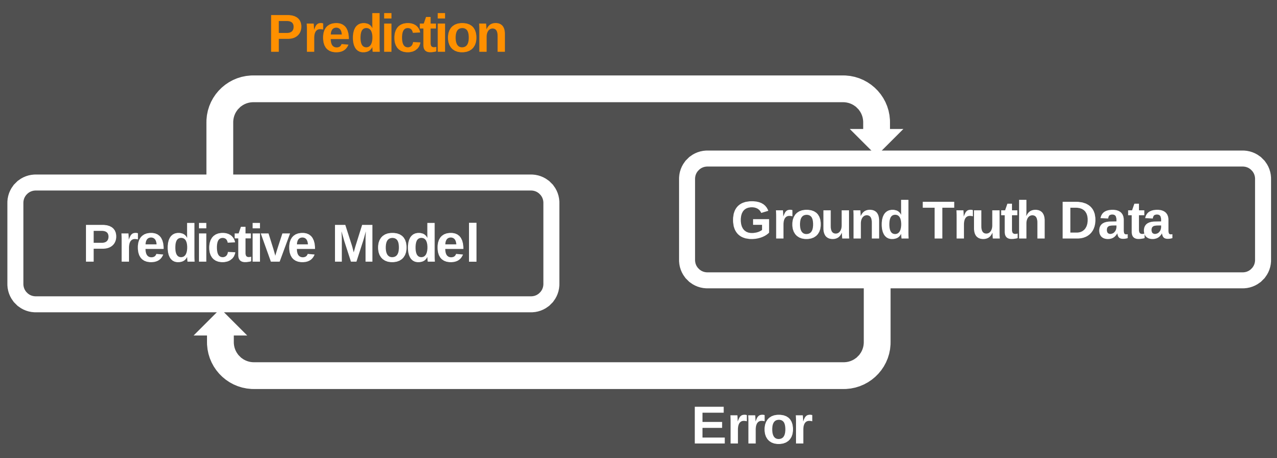


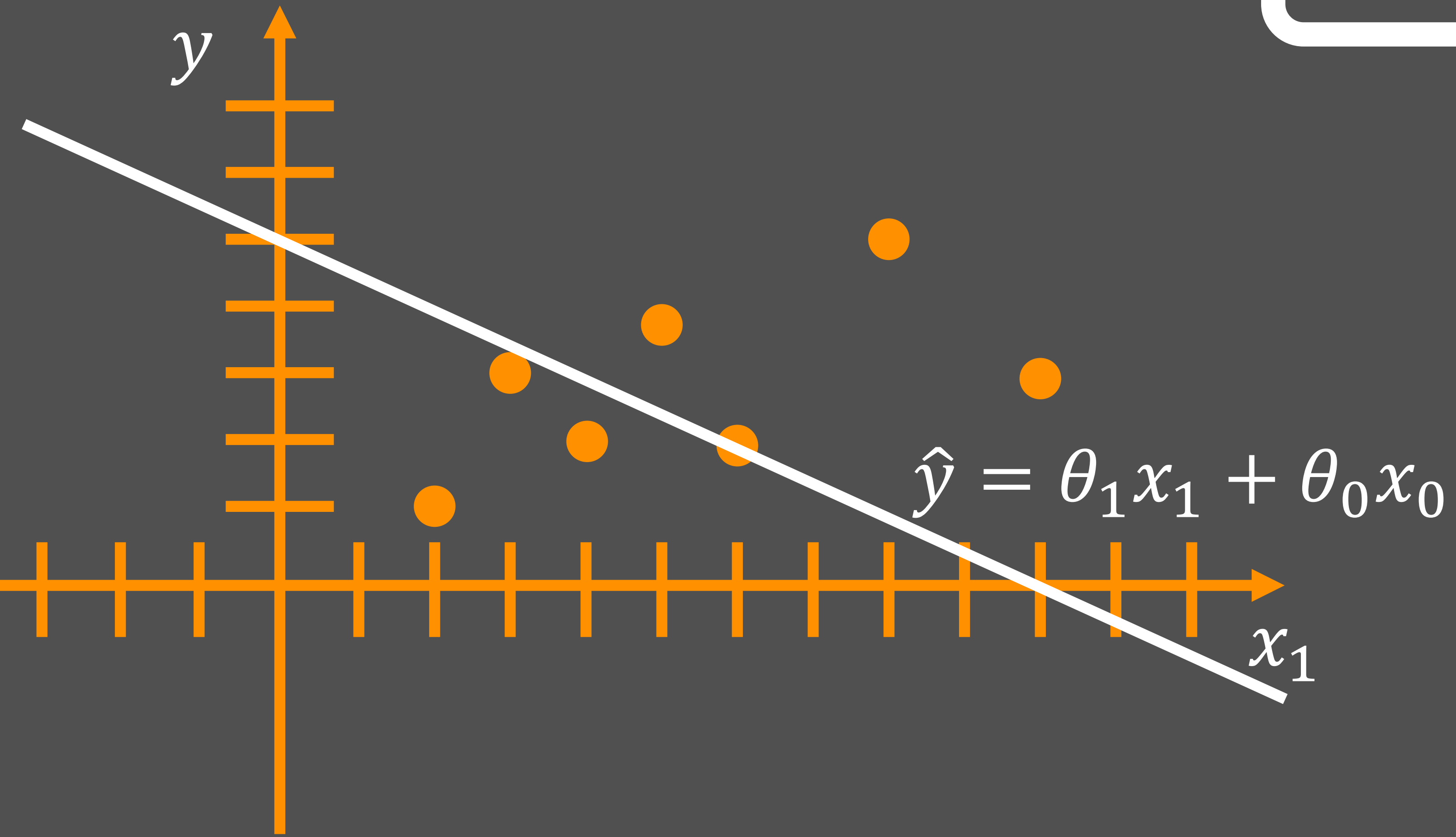


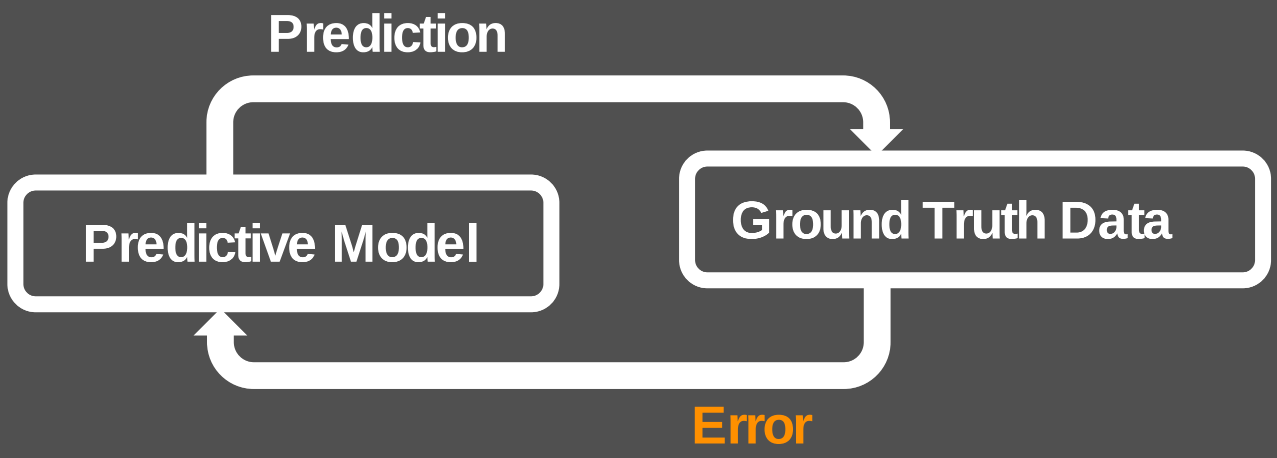
Ground Truth Data

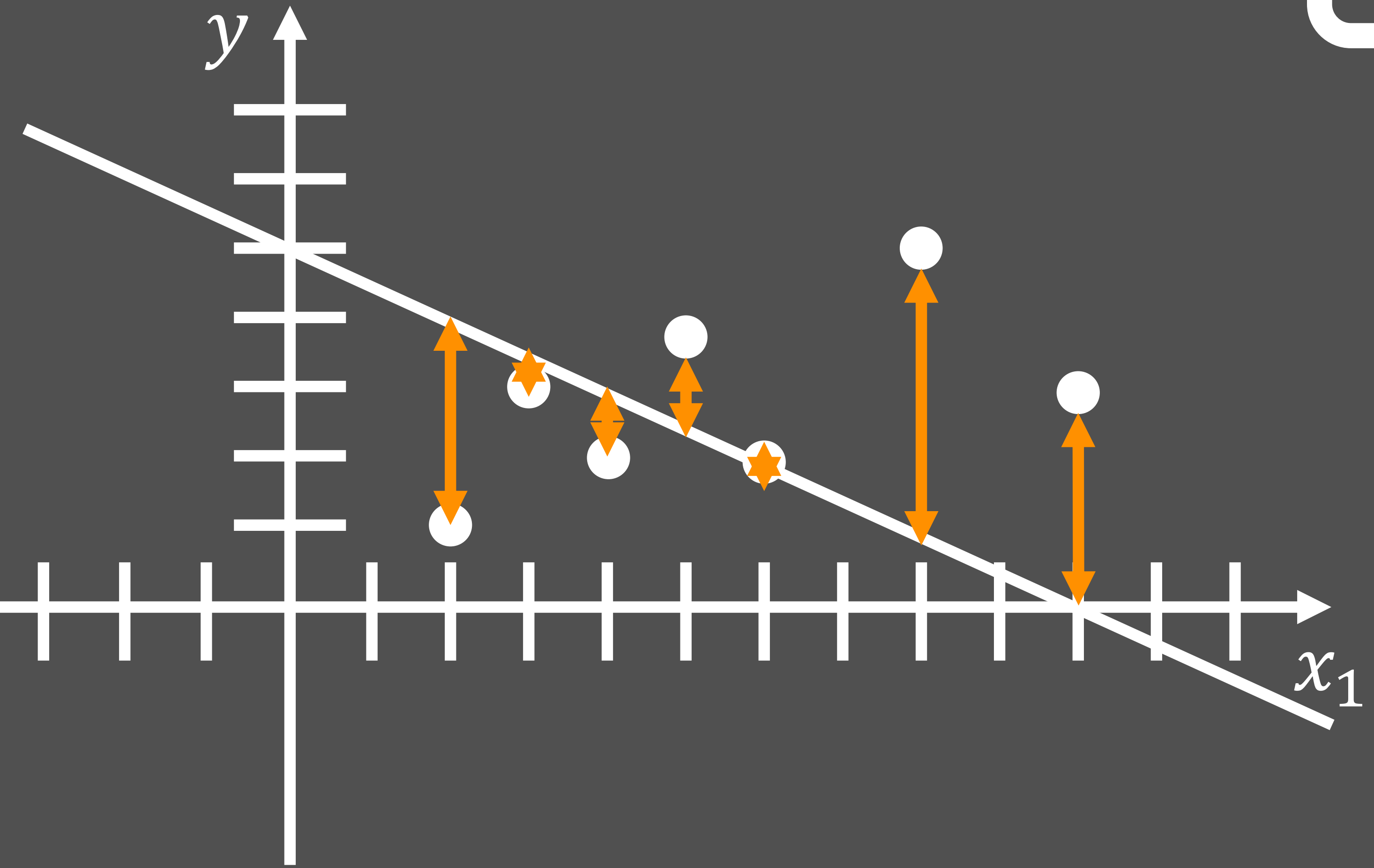
Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

Random Weights

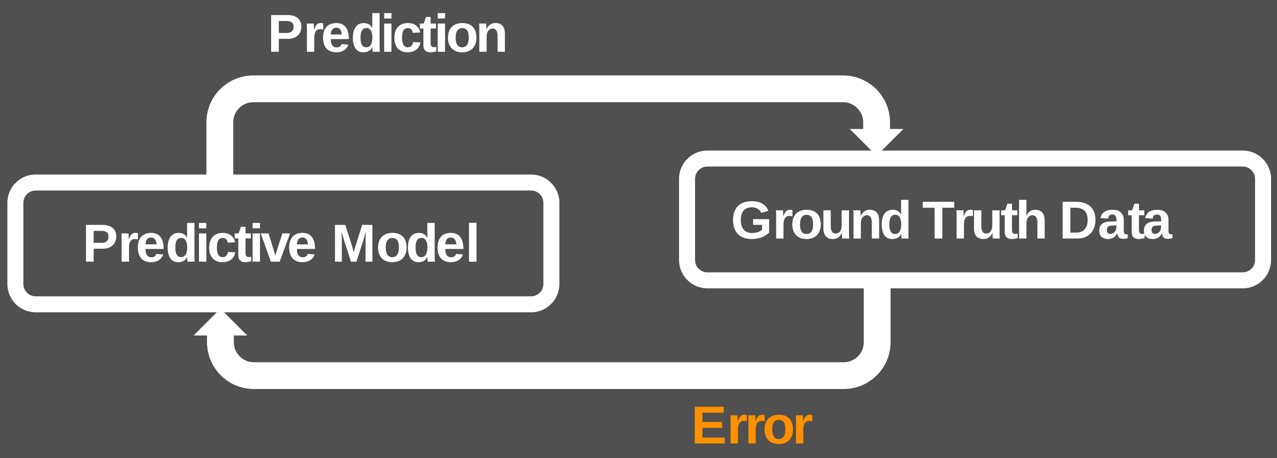


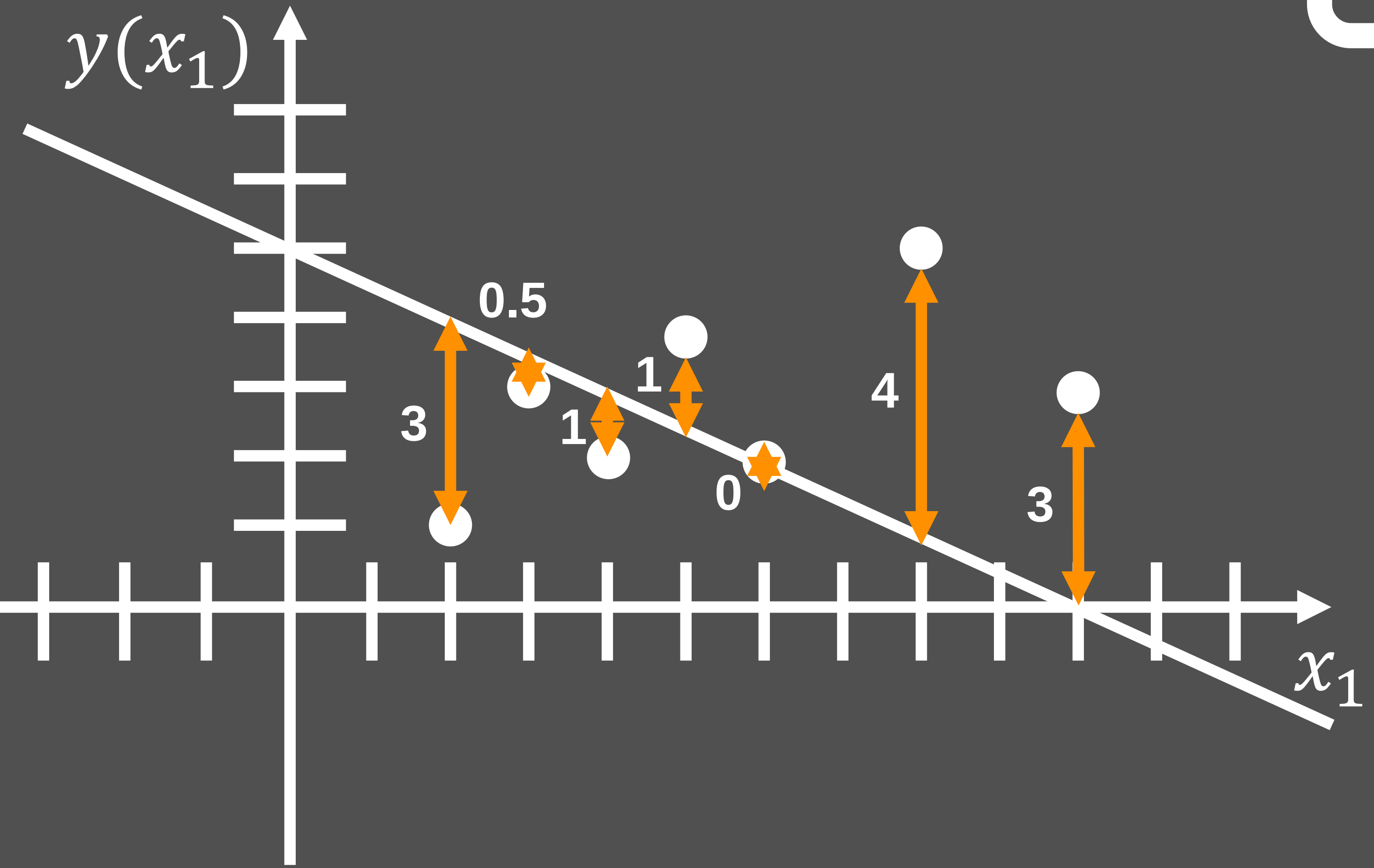






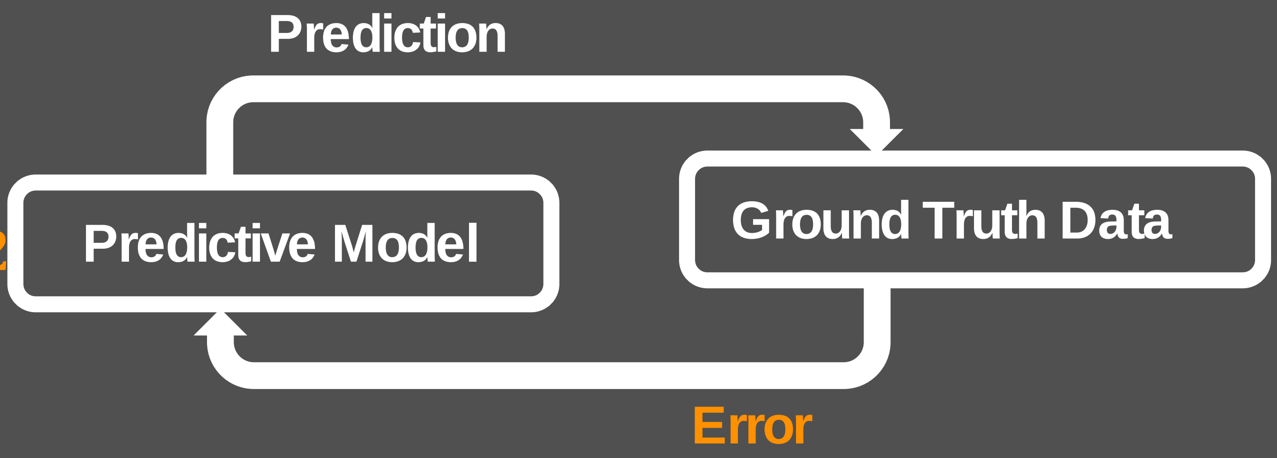
$$E = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

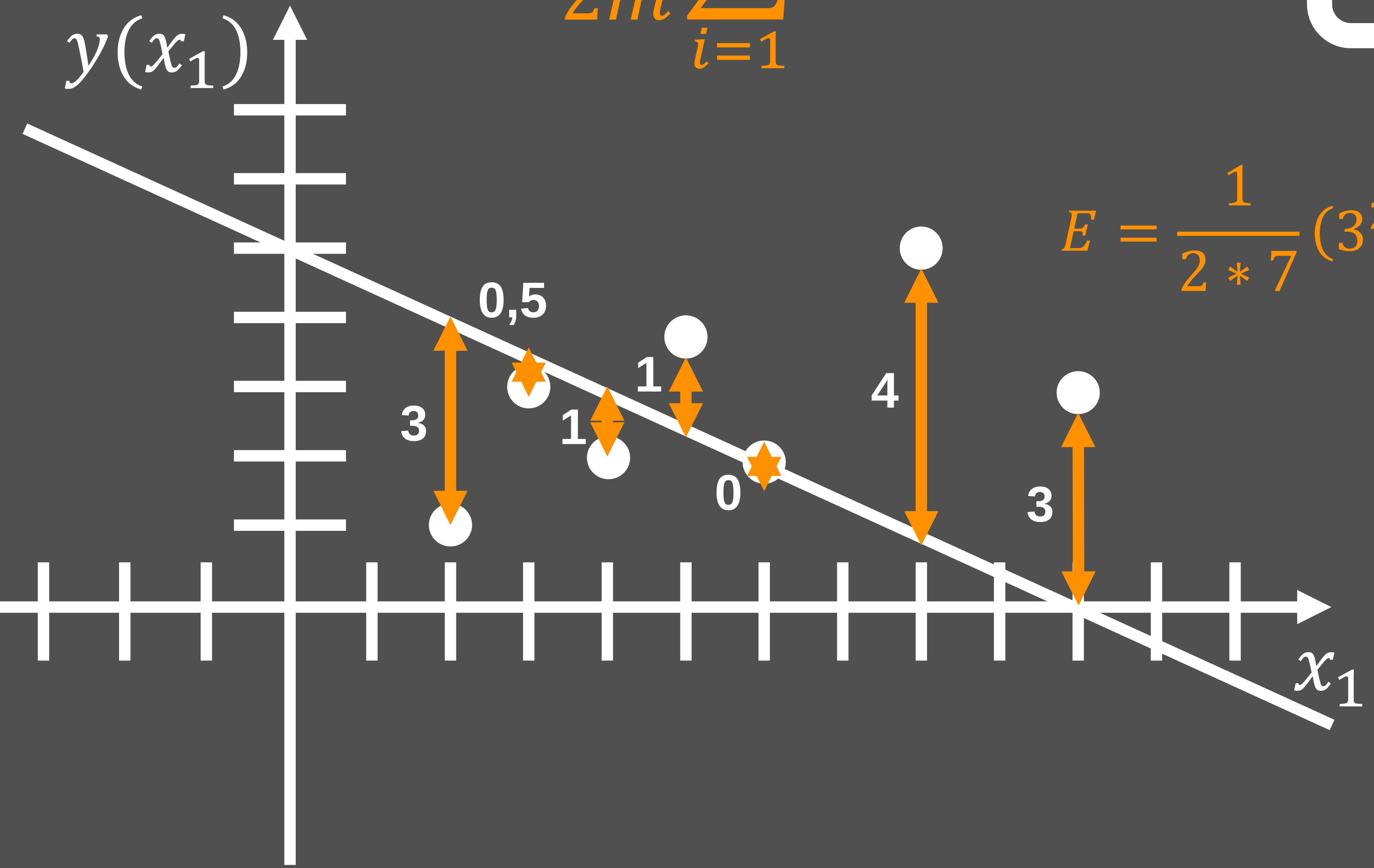




$$E = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$$

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





$$E = \frac{1}{2 * 7}(3^2 + 0{,}5^2 + 1^2 + 1^2 + 0^2 + 4^2 + 3^2)$$

$$E = 2.859$$

$$m = 7$$

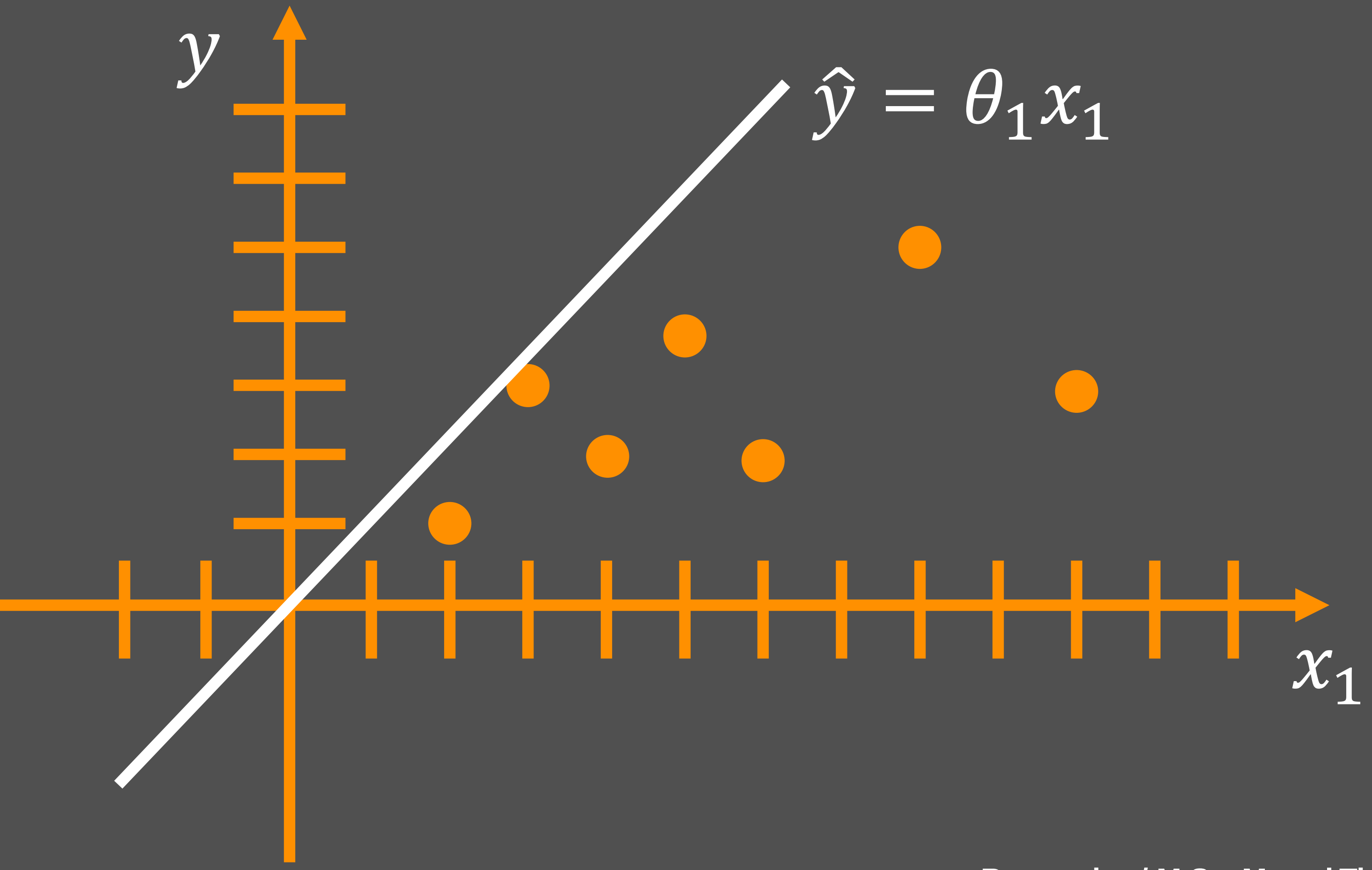
Prediction would be as good as error of 2,589

Gradient Descent

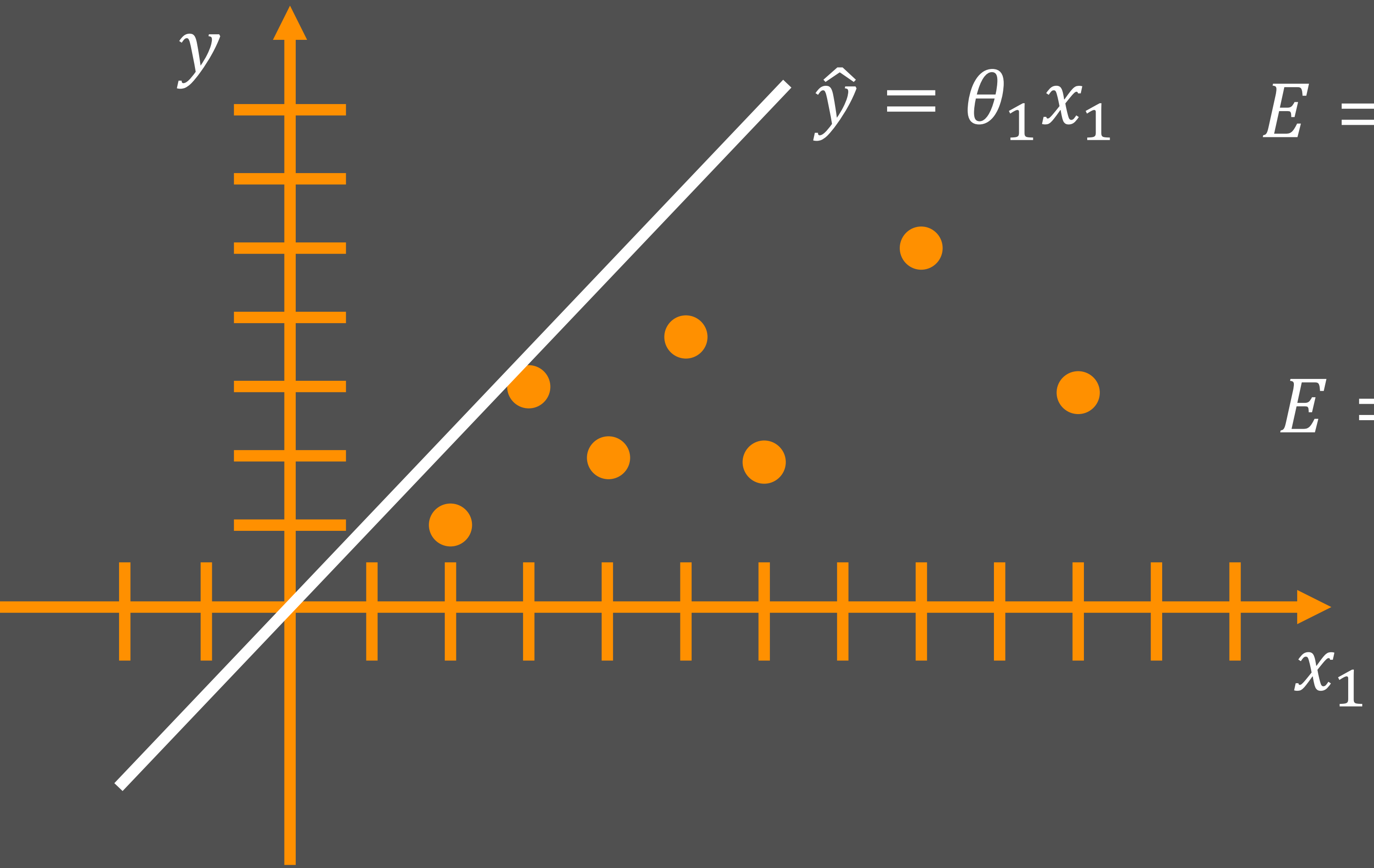
Gradient Descent:

- Tune weigths: θ_0, θ_1
- Minimize error
- Formally: $\min_{\theta_0,\theta_1} E(\theta_0, \theta_1)$
- Prediction fits better to data
- Training Process

Let‘s facilitate the problem for visualization



Gradient Descent



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

$$E = \frac{1}{2m} \sum_{i=1}^{m} ((\theta_1 x_{i,1}) - y_i)^2$$

Gradient Descent

$$E = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}((\theta_1 x_{i,1}) - y_i)^2$$

Want: $\min_{\theta_1} E(\theta_1)$

Calculate: $\frac{\partial E(\theta_1)}{\partial \theta_1} = 2 * \frac{1}{2m} * \sum_{i=1}^{m}\left(\left(\theta_1 x_{i,1}\right) - y_i\right) * x_{i,1}$

$$\frac{\partial E(\theta_1)}{\partial \theta_1} = \frac{1}{m}\sum_{i=1}^{m}\left(\left(\theta_1 x_{i,1}\right) - y_i\right) * x_{i,1}$$

Gradient Descent

Tune weight: θ_1

$$\theta_1 := \theta_1 - \alpha * \frac{\partial E(\theta_1)}{\partial \theta_1}$$

Algorithm:

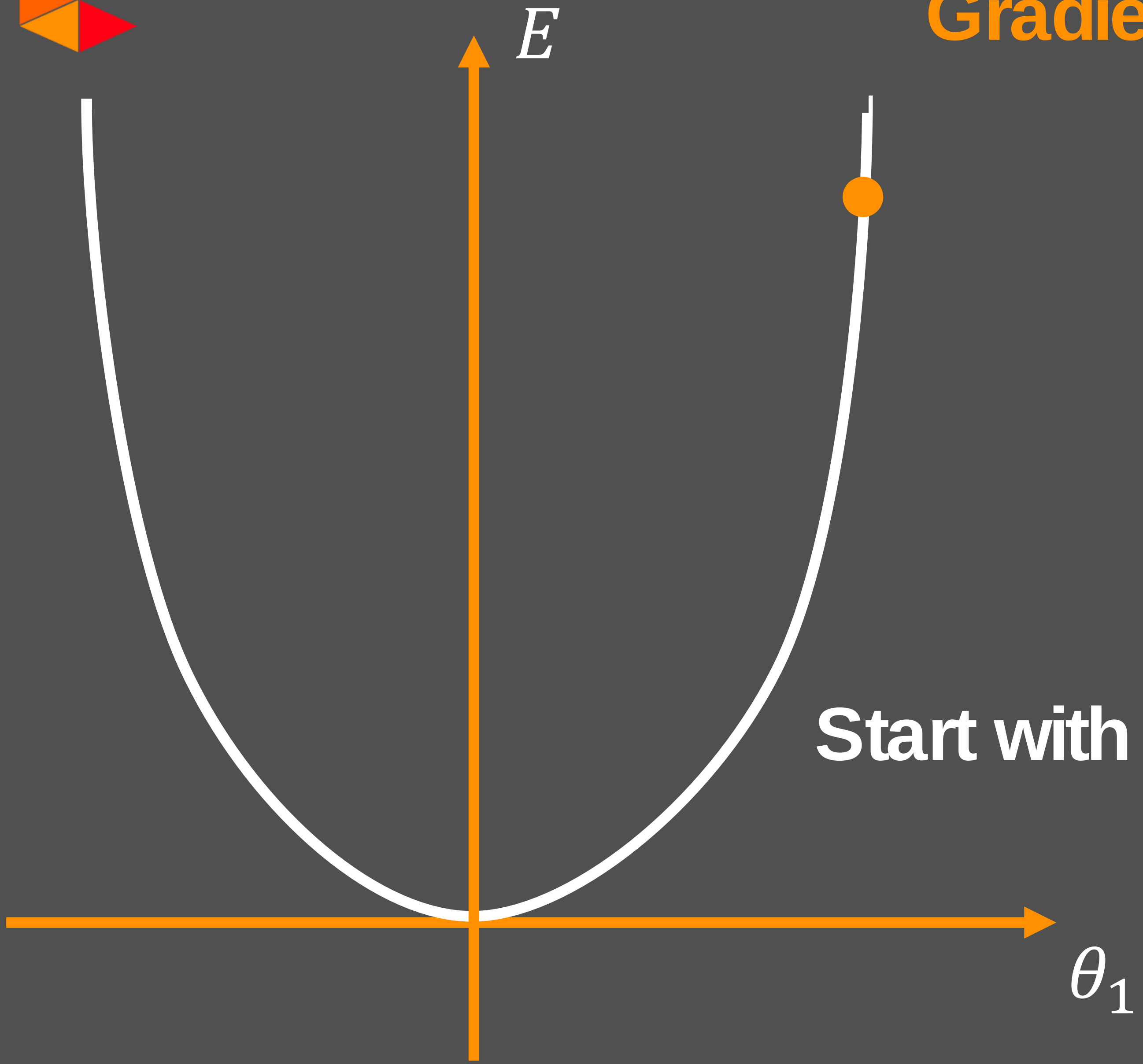
Start with random θ_1

repeat until convergence{

$$\theta_1 := \theta_1 - \alpha * \frac{\partial E(\theta_1)}{\partial \theta_1}$$

}

Gradient Descent



Start with random θ_1

Gradient Descent

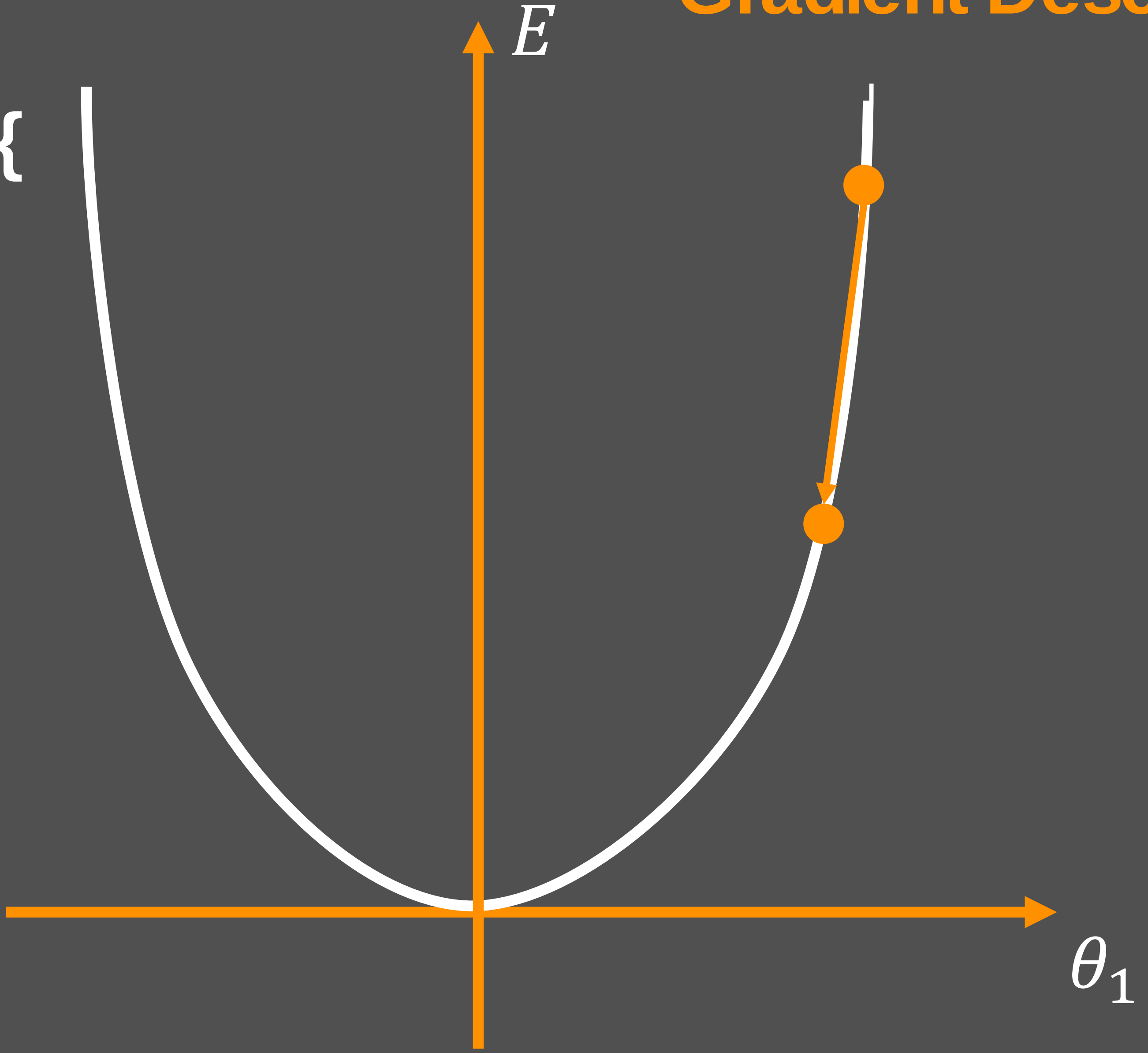
repeat until convergence{

$$\theta_1 := \theta_1 - \alpha * \frac{\partial E(\theta_1)}{\partial \theta_1}$$

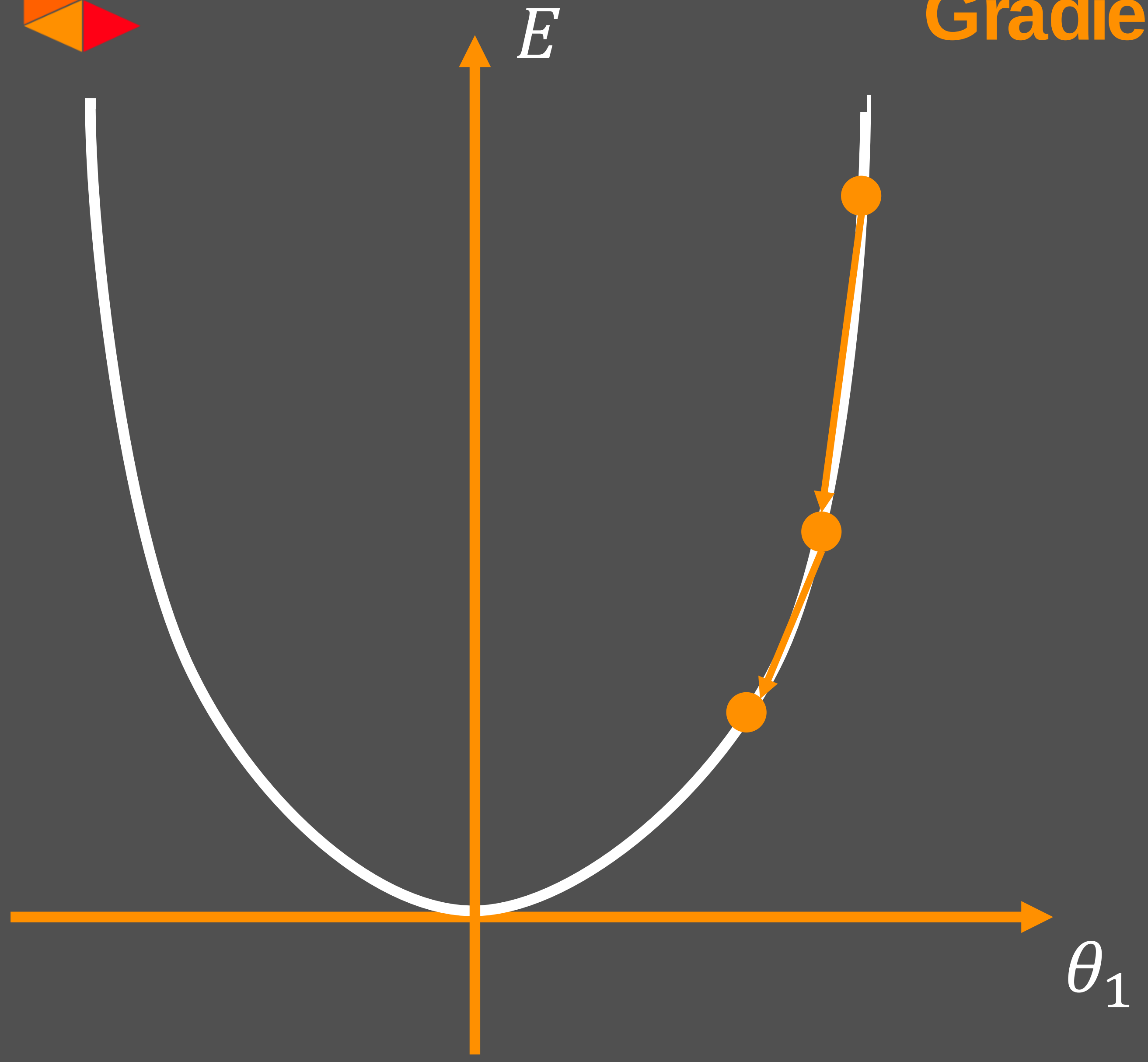
}

α: **Learning Rate**

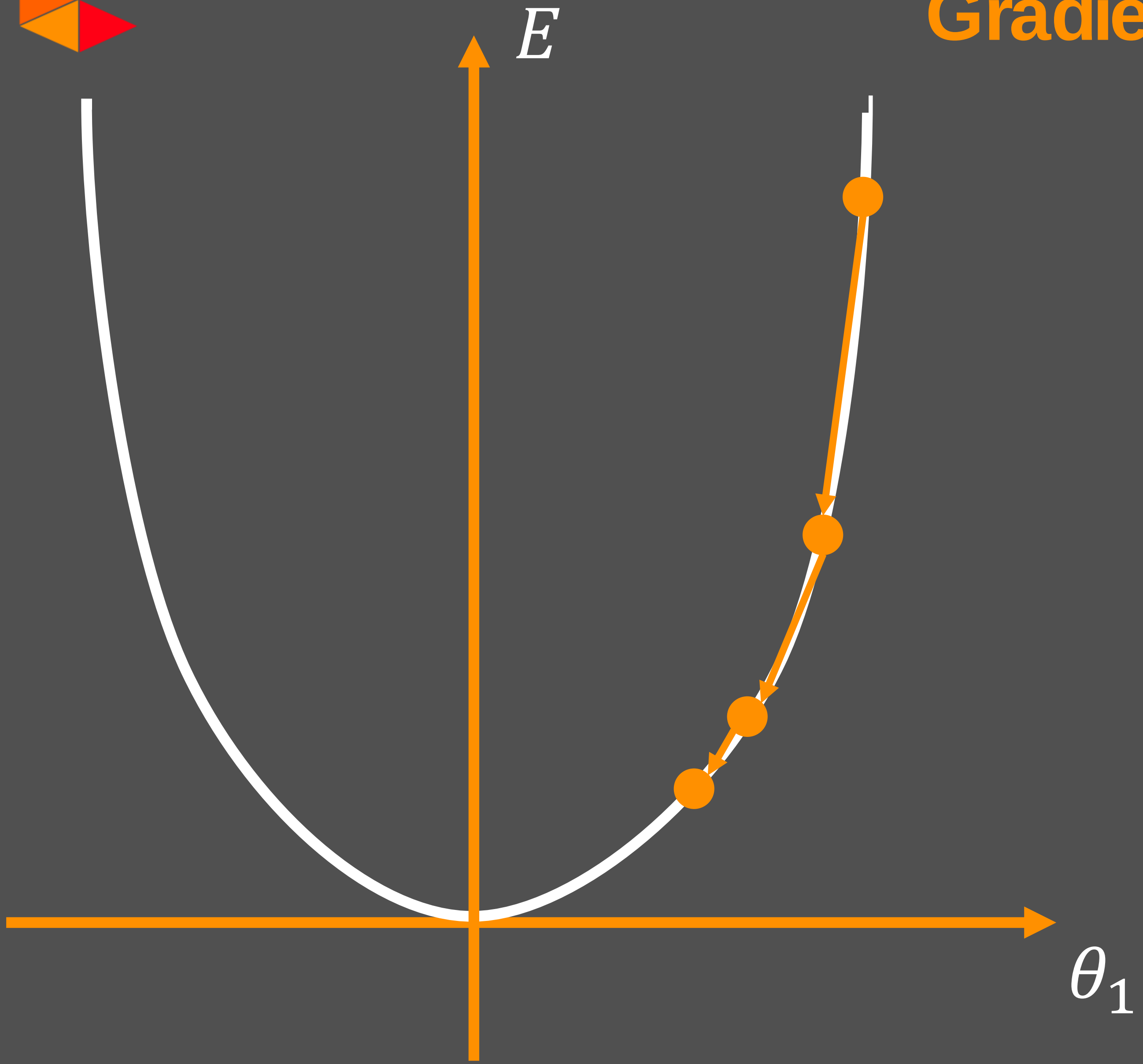
$$\alpha = const.$$



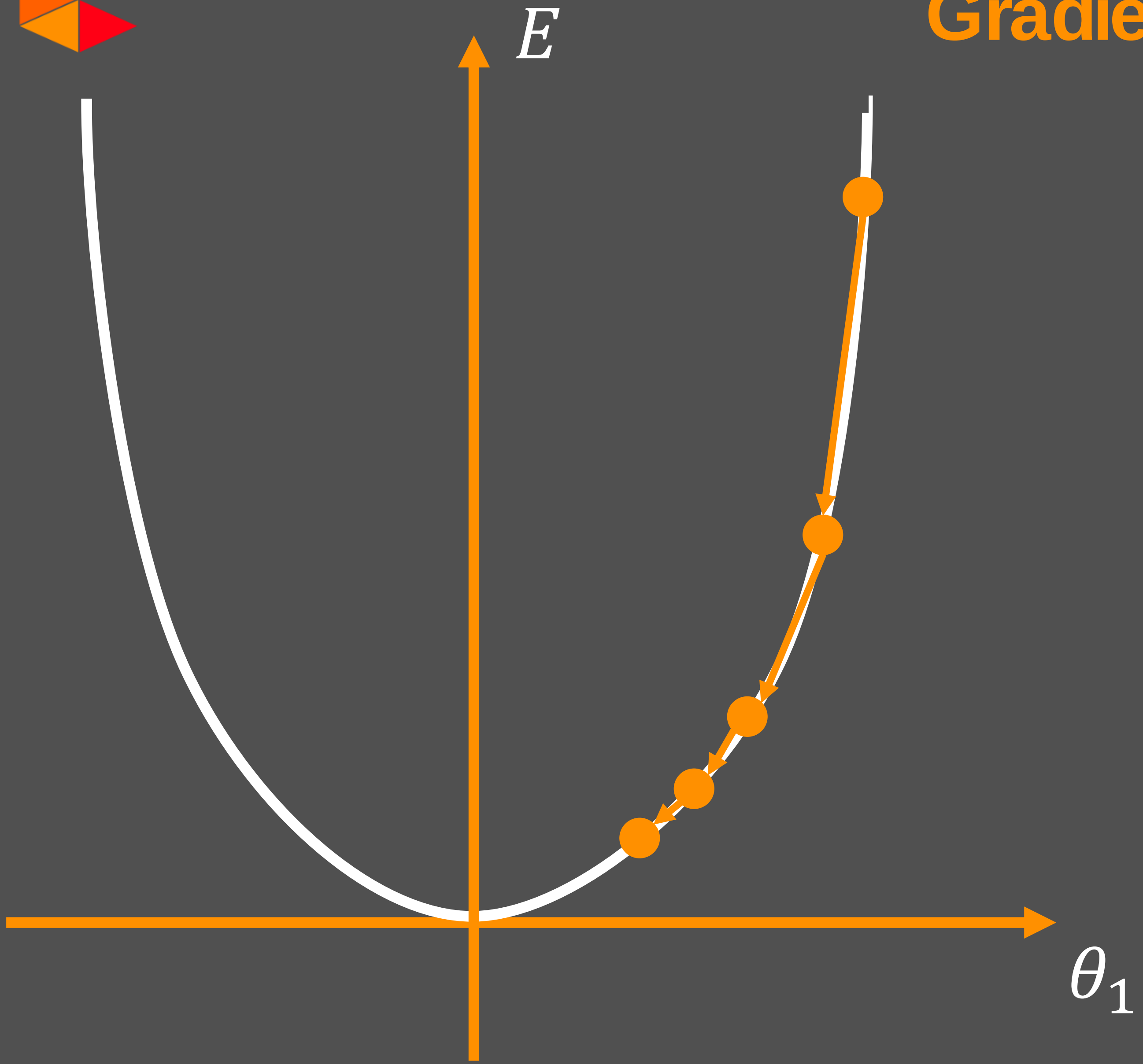
Gradient Descent



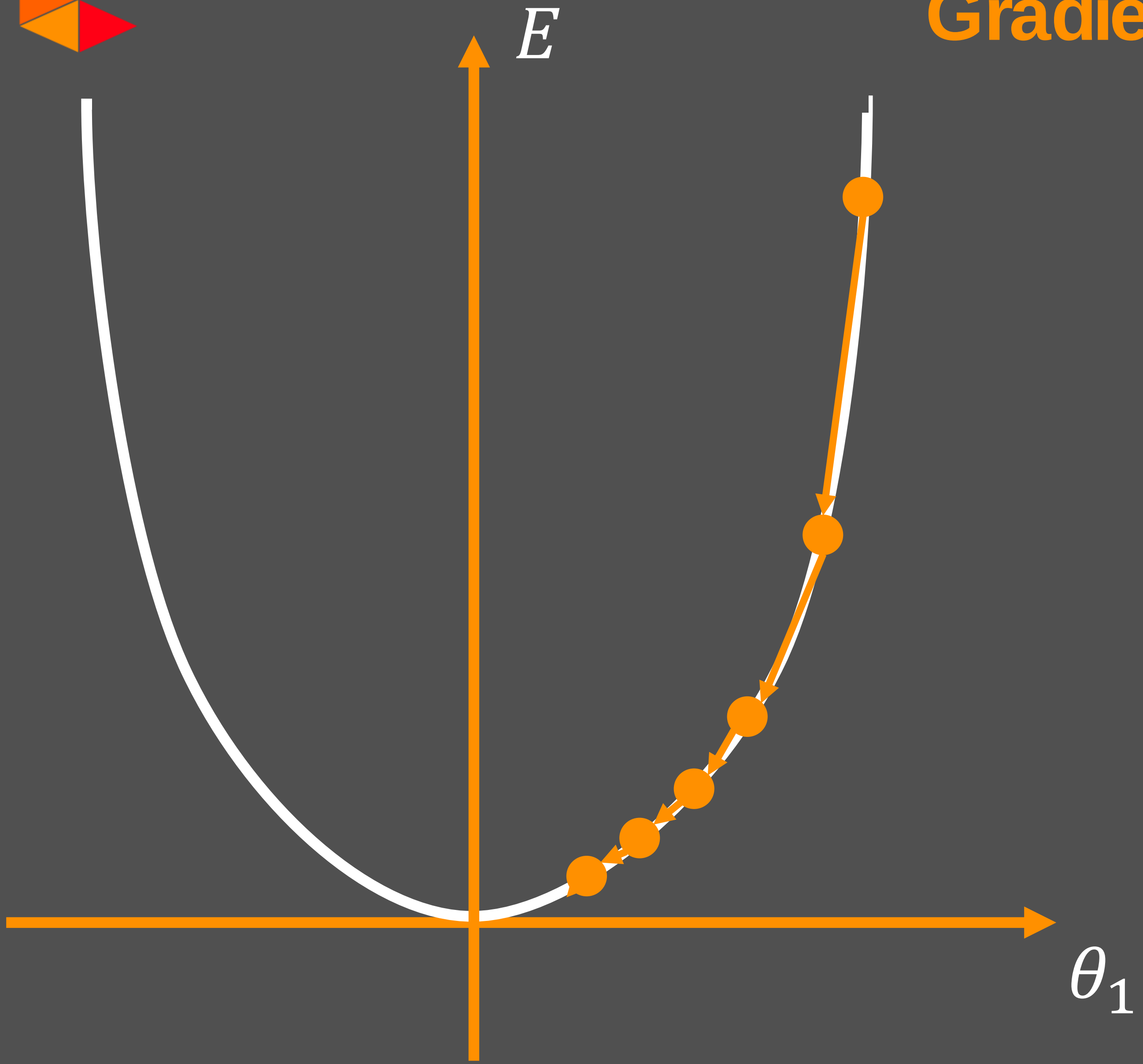
Gradient Descent



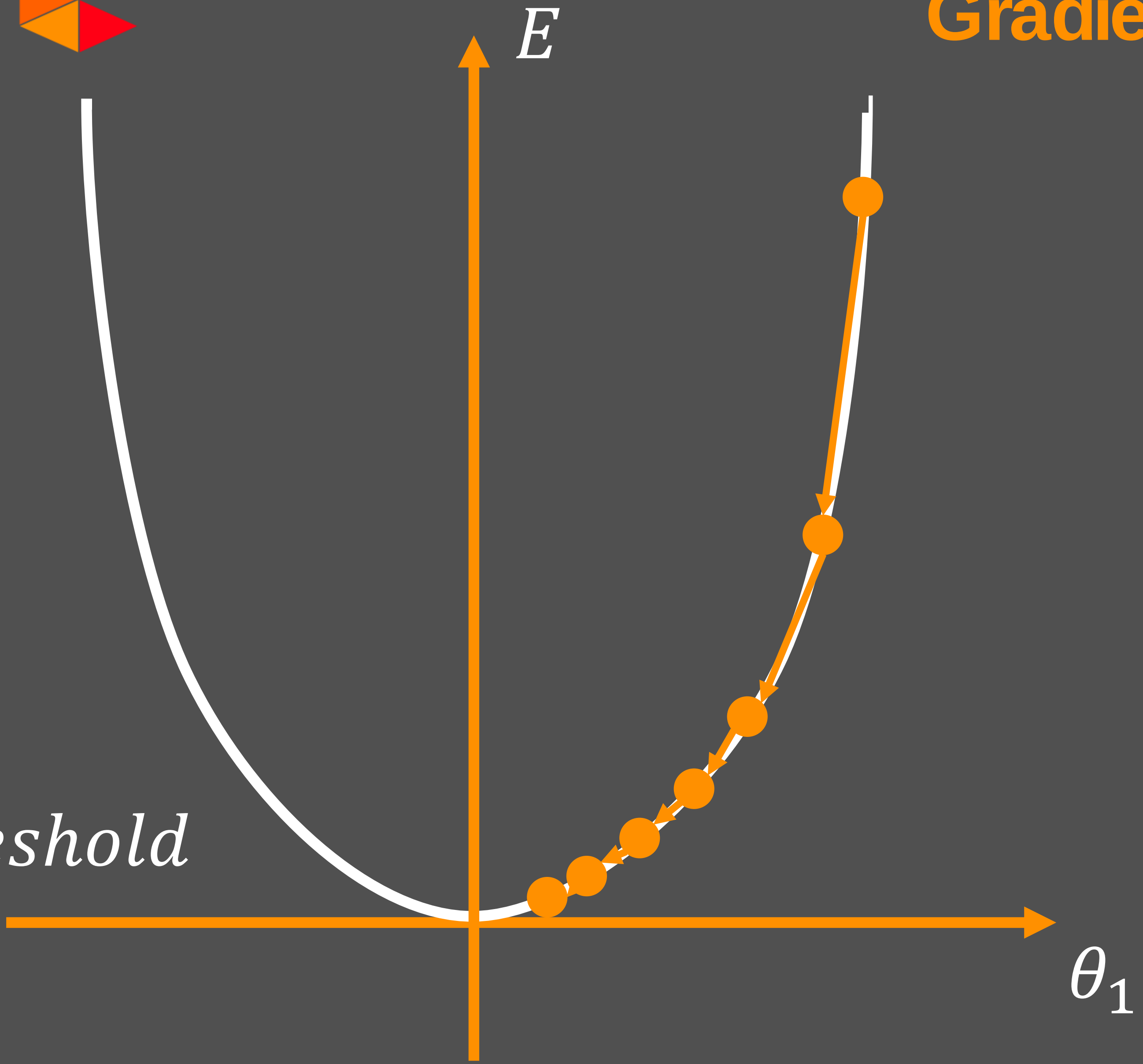
Gradient Descent



Gradient Descent



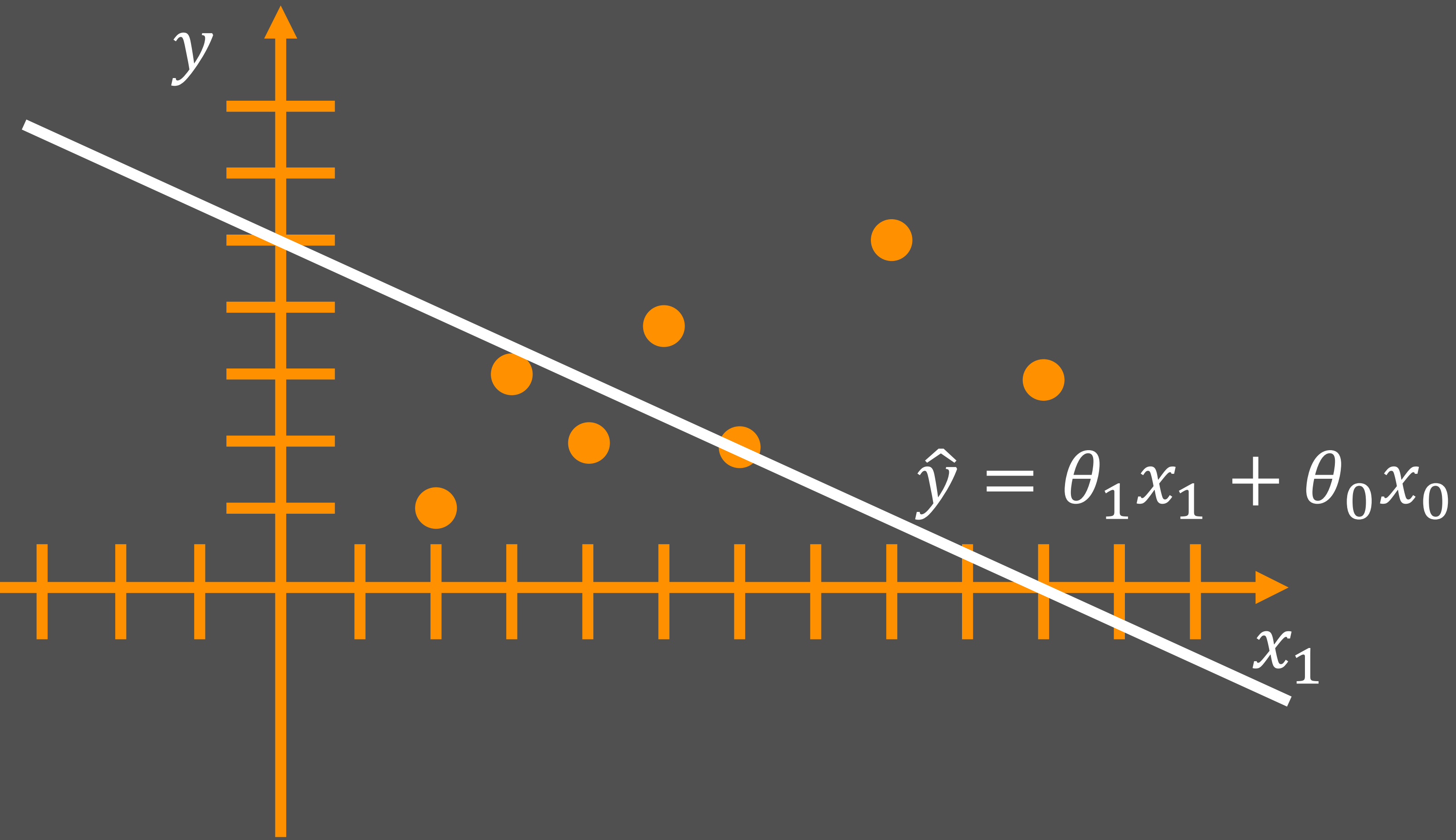
Gradient Descent



$$e.g.: E < threshold$$

Gradient Descent

Two weights:



Gradient Descent

$$E = \frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2m}\sum_{i=1}^{m}((\theta_1 x_{i,1} + \theta_0 x_{i,0}) - y_i)^2$$

Want: $\min_{\theta_0,\theta_1} E(\theta_0,\theta_1)$

$$\frac{\partial E(\theta_0,\theta_1)}{\partial\theta_1} = \frac{1}{m}\sum_{i=1}^{m}\left(\left(\theta_1 x_{i,1}\right) - y_i\right) * x_{i,1}$$

$$\frac{\partial E(\theta_0,\theta_1)}{\partial\theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(\left(\theta_0 x_{i,0}\right) - y_i\right) * x_{i,0}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(\left(\theta_0 x_{i,0}\right) - y_i\right), x_{i,0} = 1$$

Gradient Descent

Start with random θ_0, θ_1

repeat until convergence{

$$\theta_0 := \theta_0 - \alpha * \frac{\partial E(\theta_0,\theta_1)}{\partial \theta_0}$$

$$\theta_1 := \theta_1 - \alpha * \frac{\partial E(\theta_0,\theta_1)}{\partial \theta_1}$$

}

Gradient Descent

Implementation note:

Start with random θ_0, θ_1

repeat until convergence{

$$tmp_0 := \theta_0 - \alpha * \frac{\partial E(\theta_0, \theta_1)}{\partial \theta_0}$$

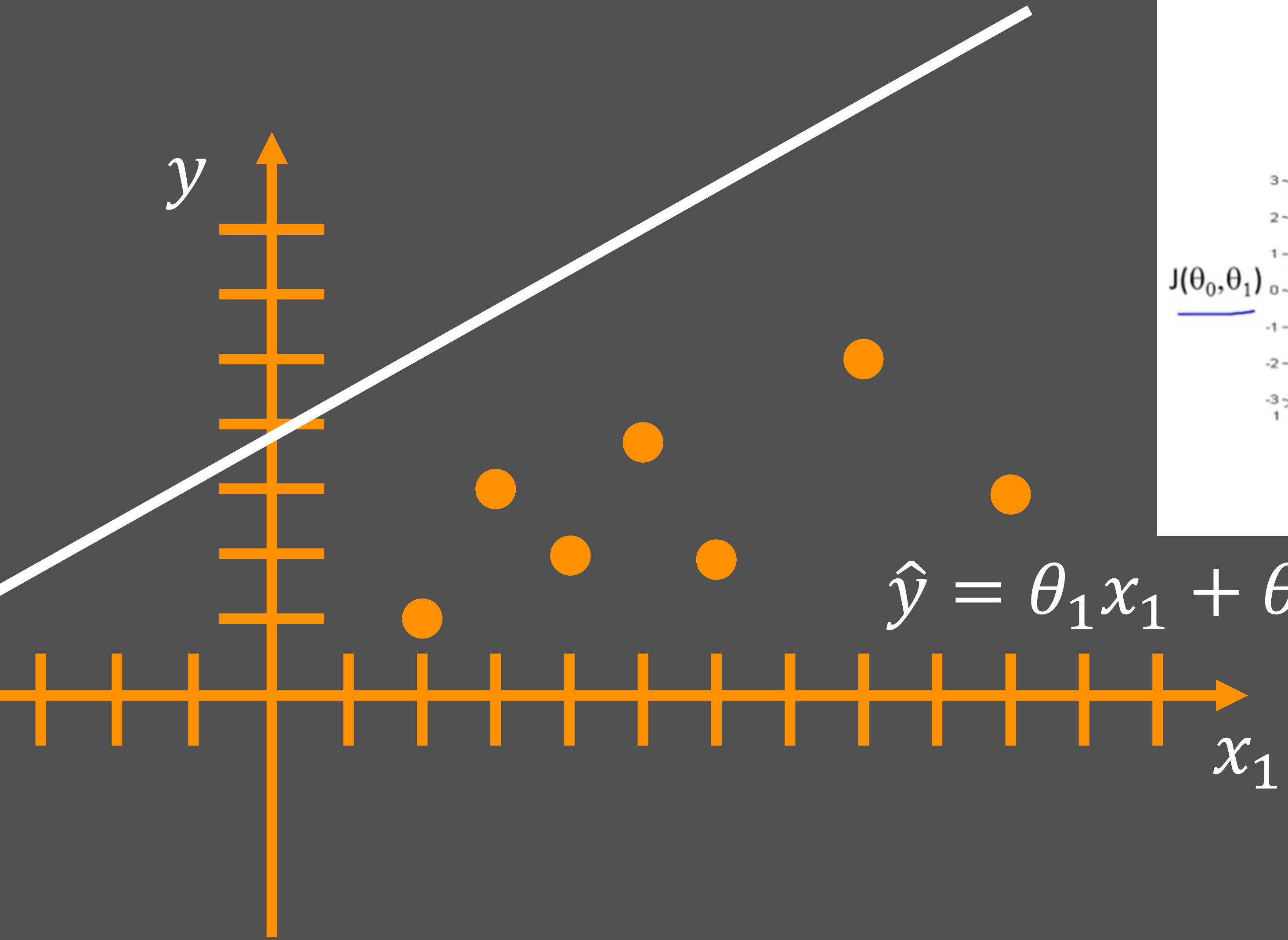
$$tmp_1 := \theta_1 - \alpha * \frac{\partial E(\theta_0, \theta_1)}{\partial \theta_1}$$

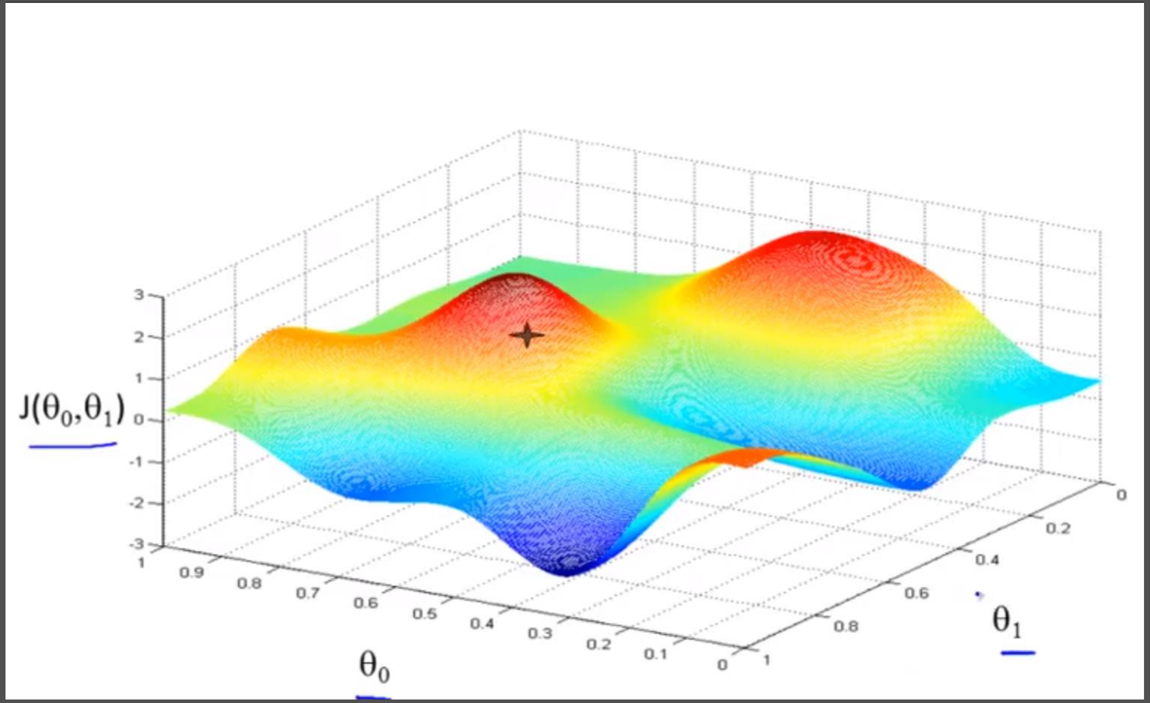
$$\theta_0 := tmp_0$$

$$\theta_1 := tmp_1$$

}

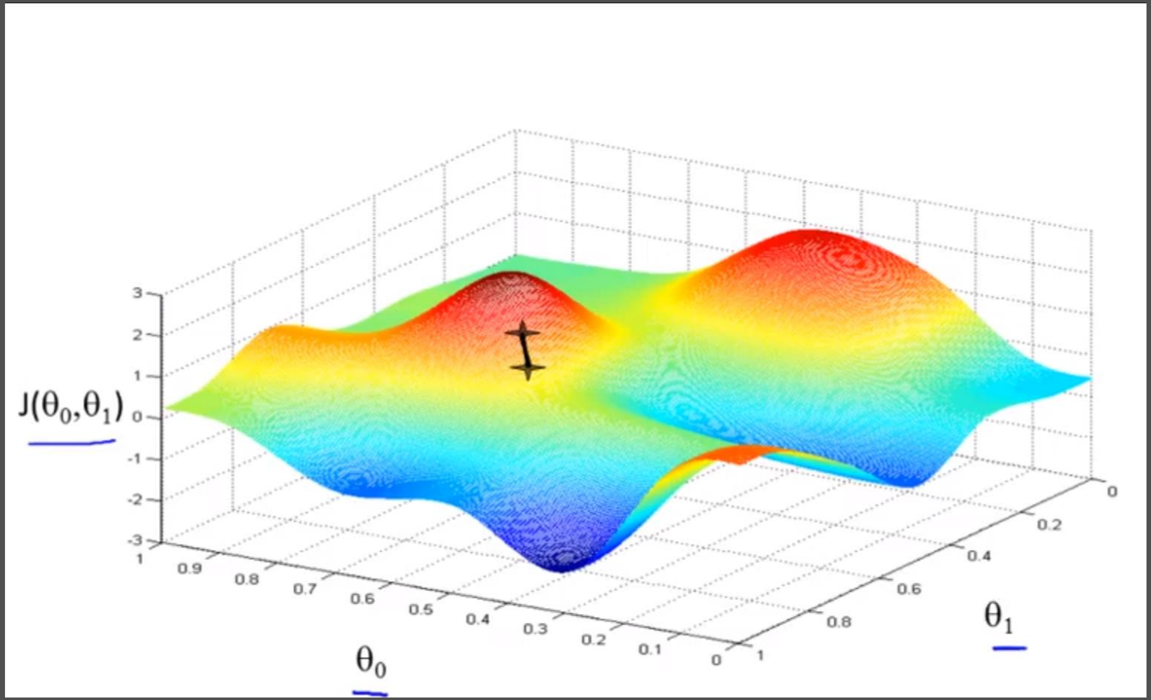
Gradient Descent

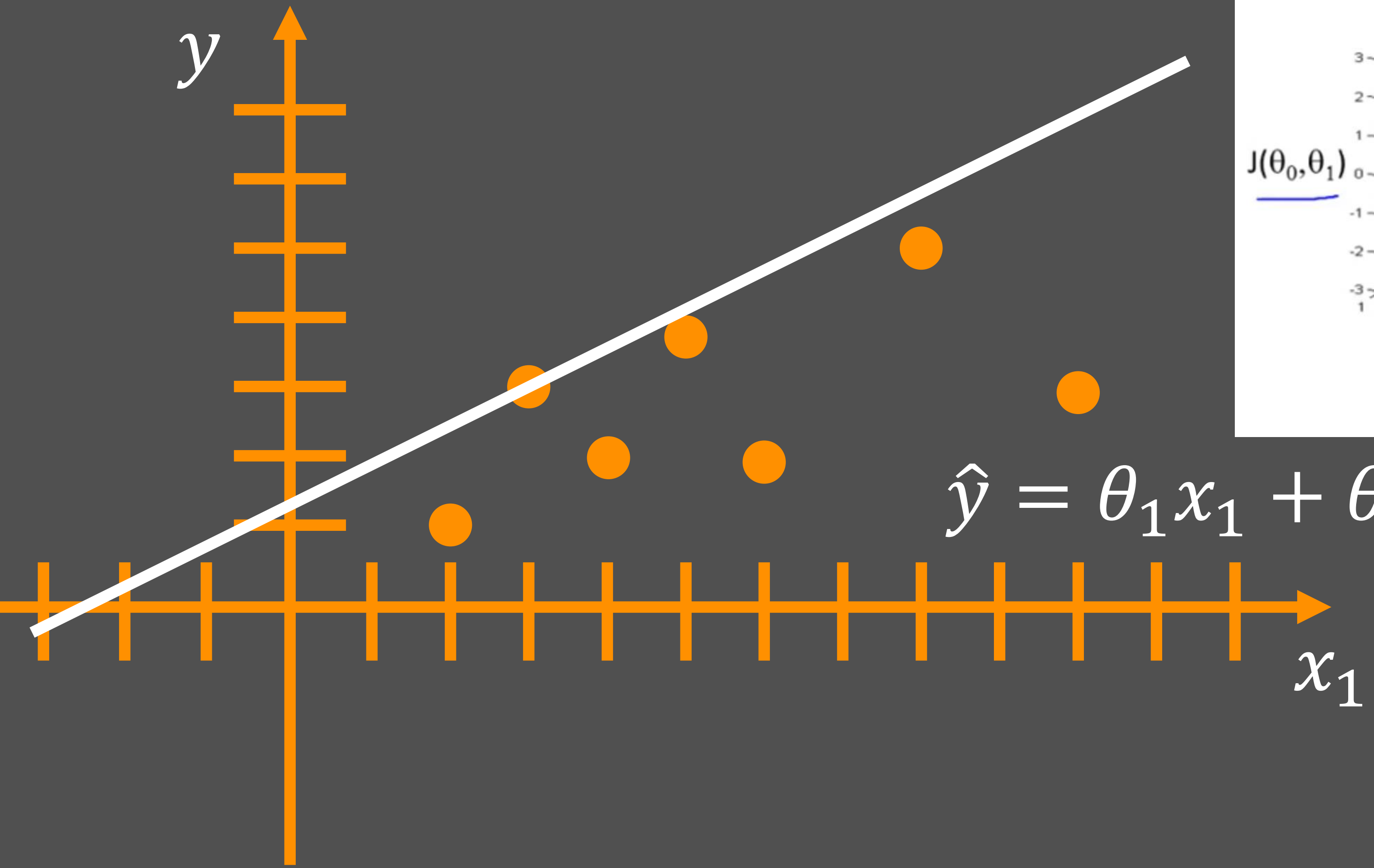




$$\hat{y} = \theta_1 x_1 + \theta_0 x_0$$

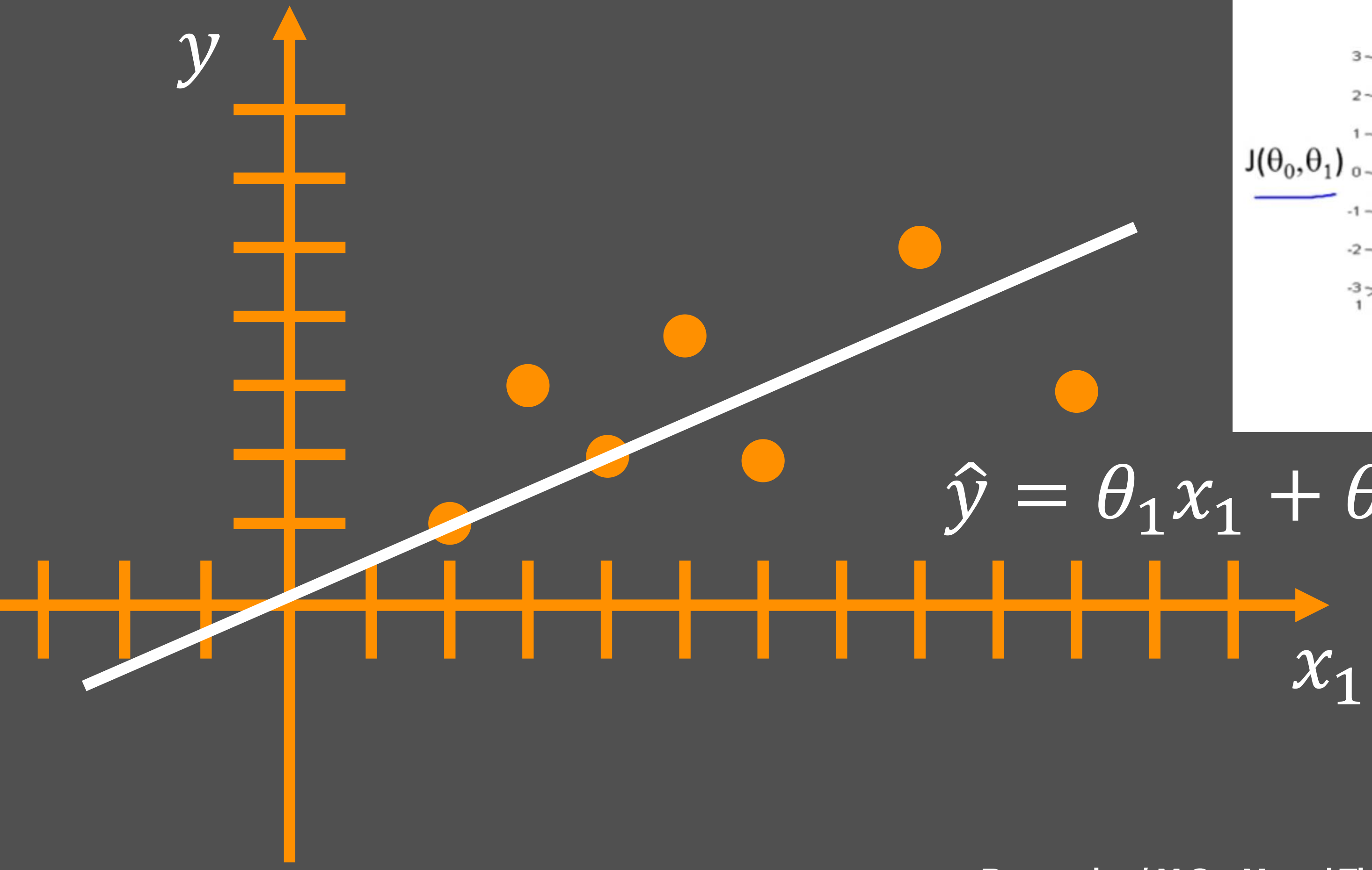
Gradient Descent

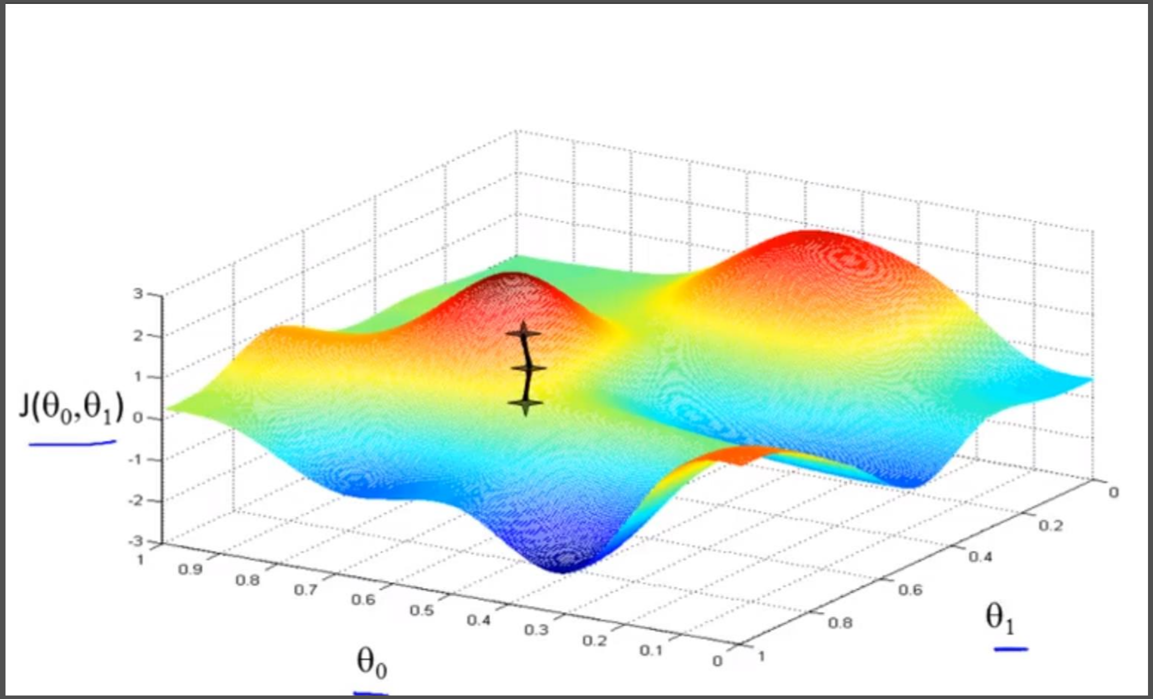




$$\hat{y} = \theta_1 x_1 + \theta_0 x_0$$

Gradient Descent

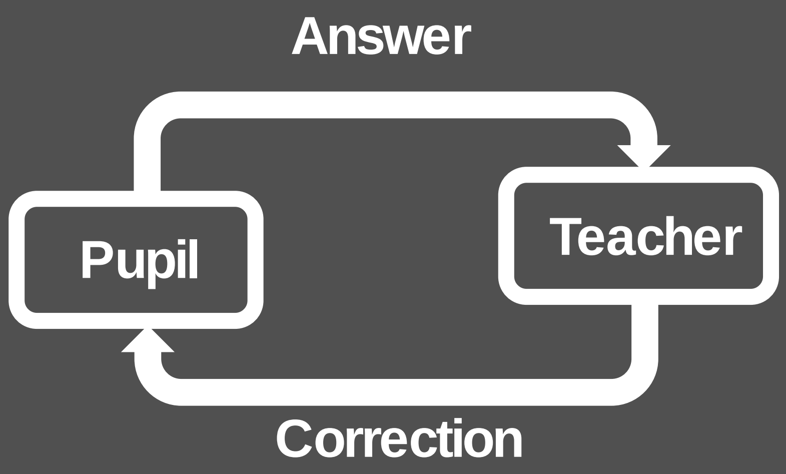


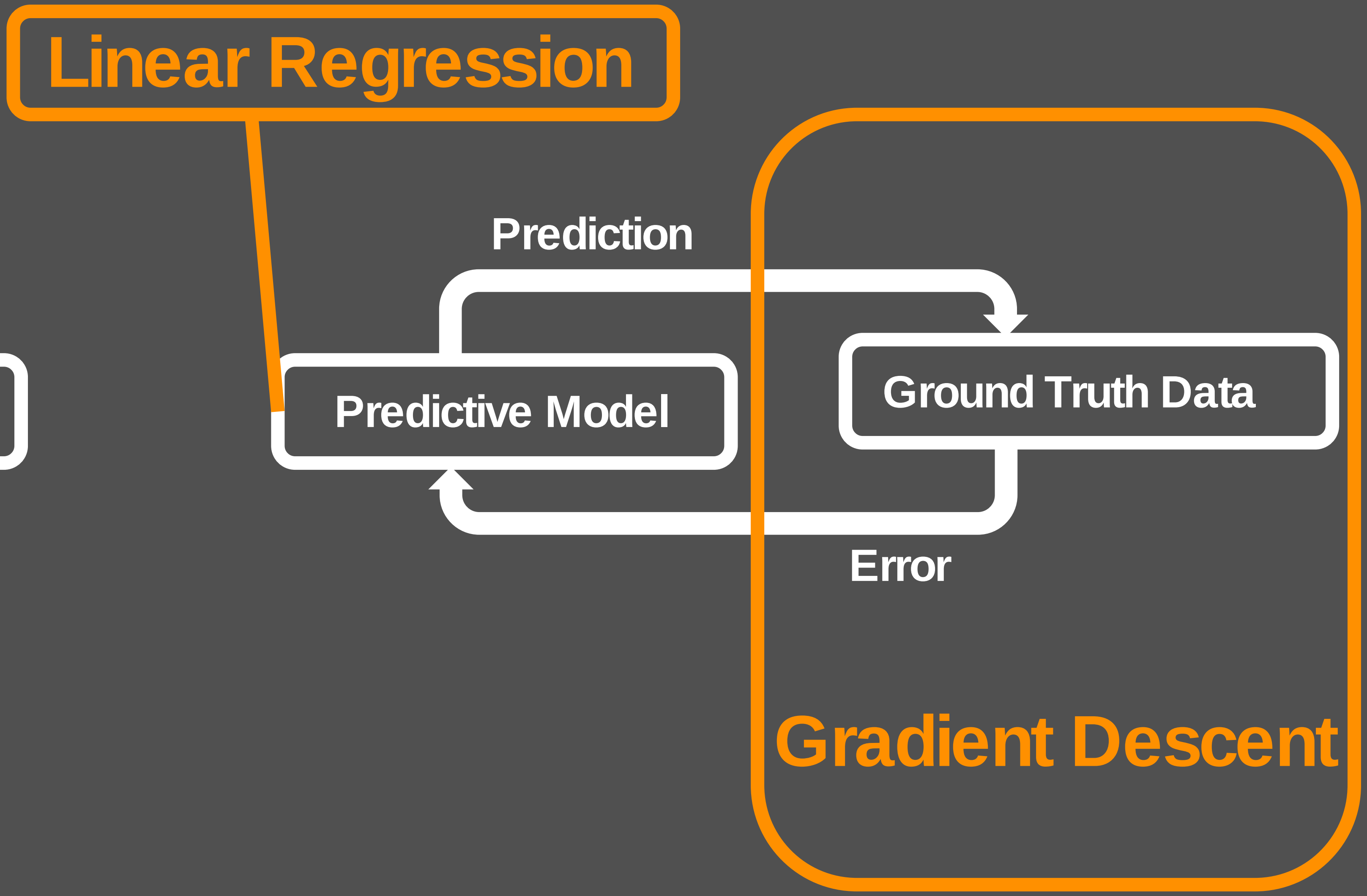


$$\hat{y} = \theta_1 x_1 + \theta_0 x_0$$

Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

You learned:





Questions?

TensorFlow

About:

- open source software library for high performance numerical computation
- variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices
- used across many other scientific domains

airbnb

AMD

NVIDIA.

UBER

SAP

kakao

DeepMind

Dropbox

ebay

Google

intel

Coca-Cola

mi

QUALCOMM

360

3DR

AIRBUS
DEFENCE & SPACE

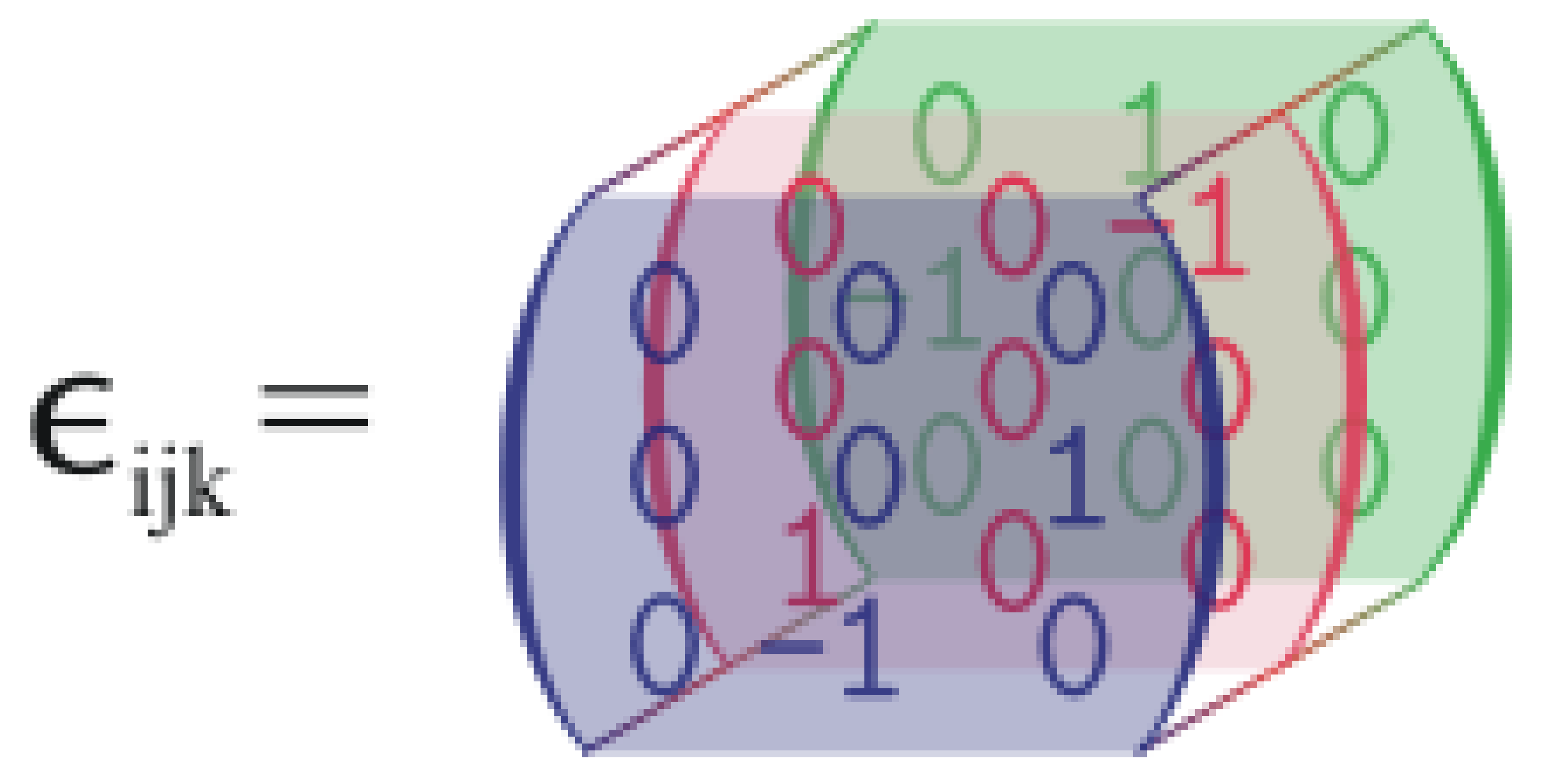
ARM

caicloud

CastBox

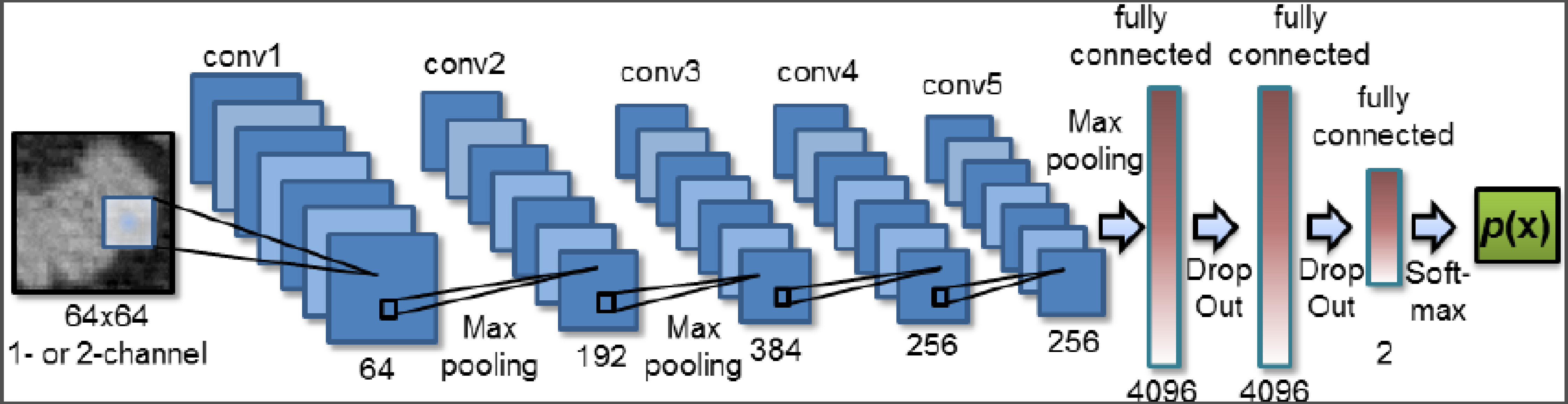
CEVA

What is a Tensor? (compare with array of array of array …)



Why is this useful?

- Try to describe multiple images in one mathematical structure



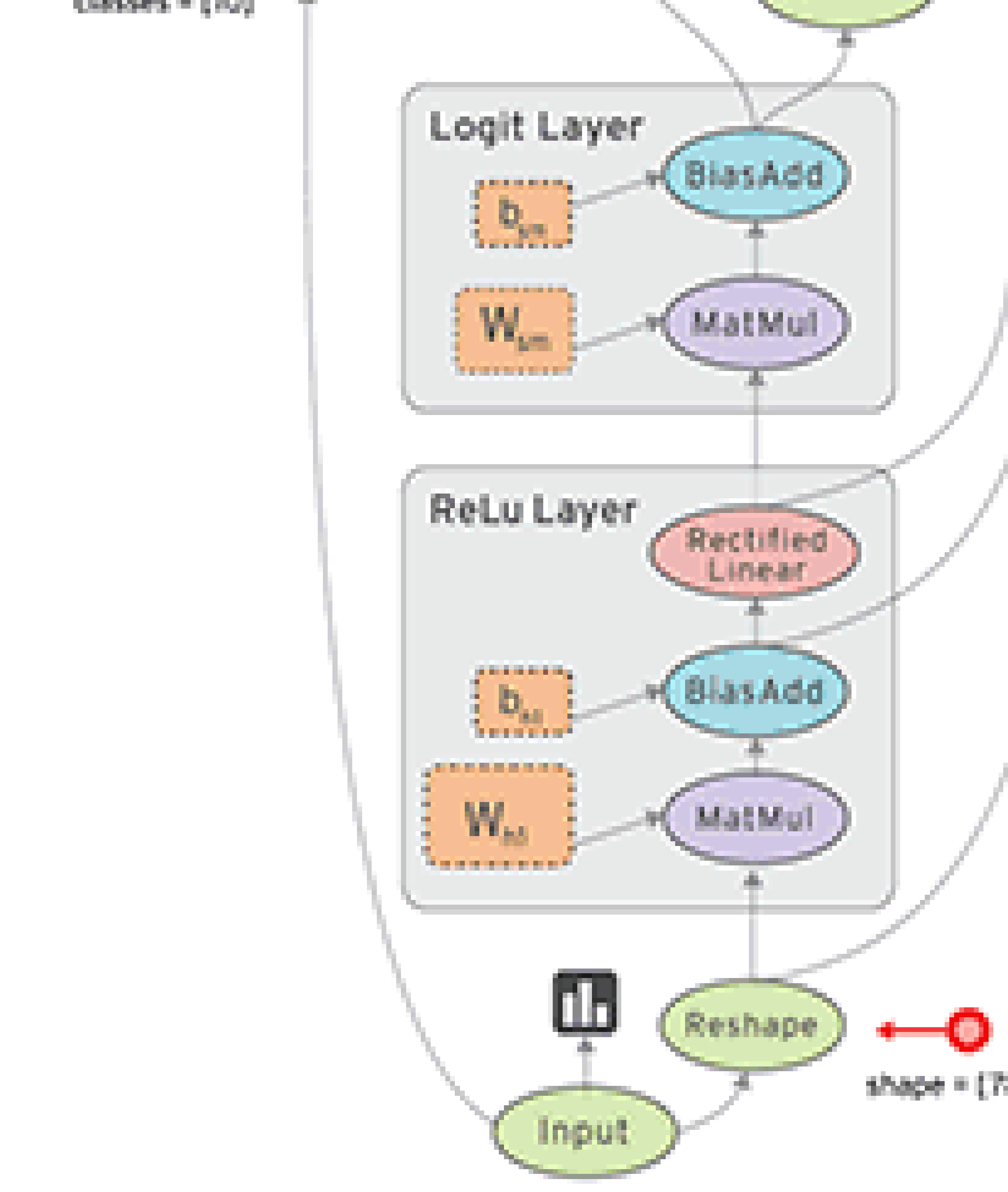
Basic elements:

- tf.Variable: remember θ
- tf.constant
- tf.placeholder: variable which can be used in a graph
- tf.flags: Global parameter, e.g. α:learning rate

Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

Graph:

- Blueprint of computations
- Chain of functions



Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

tf.placeholder:

- Useful for input data

tf.Variable:

- Values are manipulated while training

Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

Basic operations:

- tf.multiply(x, y): $x * y$ element-wise
- tf.reduce_sum(x): Computes the sum of elements across dimensions of a tensor
- tf.losses.mean_squared_error($y, \hat{y}$): $\frac{1}{2m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$
- tf.add(x, y): $x + y$
- tf.scalar_mul(α, x): $\alpha * x$

Basic operations:

- tf.transpose(x): x^T
- tf.subtract(x, y): $x - y$
- tf.reshape(X,shape): resizes X to new shape

Task today:

- Implement linear regression with gradient descent
- Execute „lr_train_gd.py“
- Implement functions in „lr_model_gd.py“
- Some helper functions „split_dataset.py“

„lr_train_gd.py“:

- Load training data
- Call train function of model
- Load test data
- Test model

Hochschule Düsseldorf
University of Applied Sciences
Fachbereich Medien
Faculty of Media

„lr_model_gd.py“:

- inference: prediction $\hat{y} = \theta_1 x_1 + \theta_0 x_0$
- loss: computation of $\mathrm{E} = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$
- gradient_descent: Computation of new θ
- train: One training step

Hints:

- n: #Columns = #Features
- $\hat{y} = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0 x_0$
- $\hat{y} = \theta x$
- Use print! For instance: print shape of tensor (tf.shape)

Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media

MIREVI

https://github.com/mati3230/modalg181