IESTI01 - TinyML

The Machine Learning Paradigm

Prof. Marcelo Rovai

May 12th, 2021

The Machine Learning Paradigm

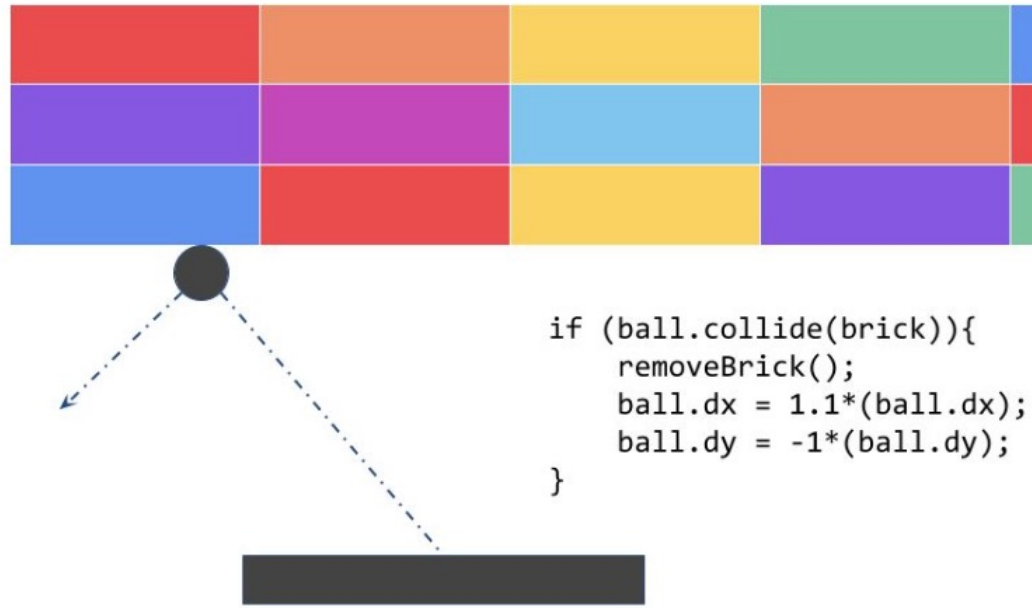
From coding to learning…

Explicit Coding

Defining rules that determine behavior of a program

Everything is pre-calculated and pre-determined by the programmer

Scenarios are limited by program complexity



```
if (ball.collide(brick)){
    removeBrick();
    ball.dx = 1.1*(ball.dx);
    ball.dy = -1*(ball.dy);
}
```

The Traditional Programming Paradigm

Consider Activity Detection

```
if(speed<4){
    status=WALKING;
}
```

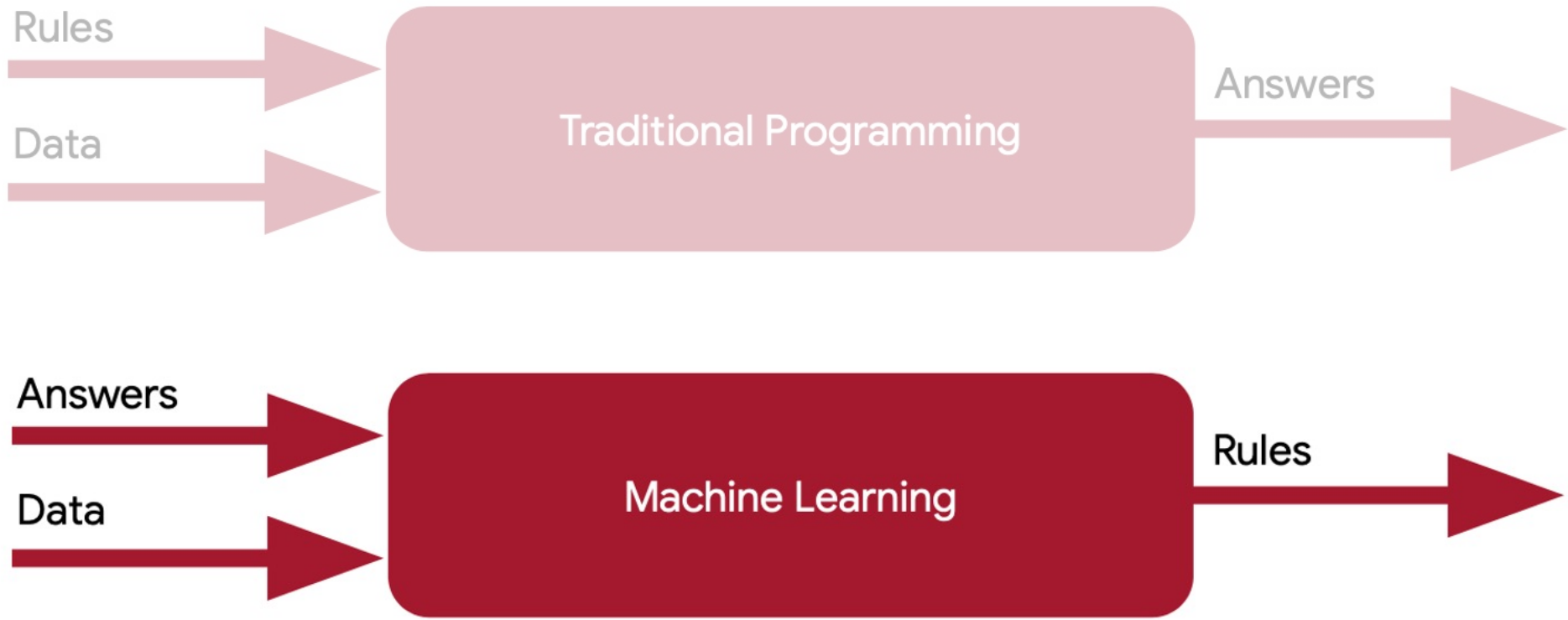
```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// ???
```

The Traditional Programming Paradigm

The Traditional Programming Paradigm



Activity Detection with Machine Learning

```
01010010101001010101
10010101010010111101
01001010100101010010
01010010101001010101
```

Label = WALKING

```
10101001010010101010
01010100100100100010
00100111110101011111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011100
10101011110101010110
11111100011110101010
```

Label = BIKING

```
11111111110100111010
00111110101111101010
01011101010101011100
10101010101001111100
```

Label = GOLFING

The Machine Learning Paradigm

```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING

```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING

```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

Label = BIKING

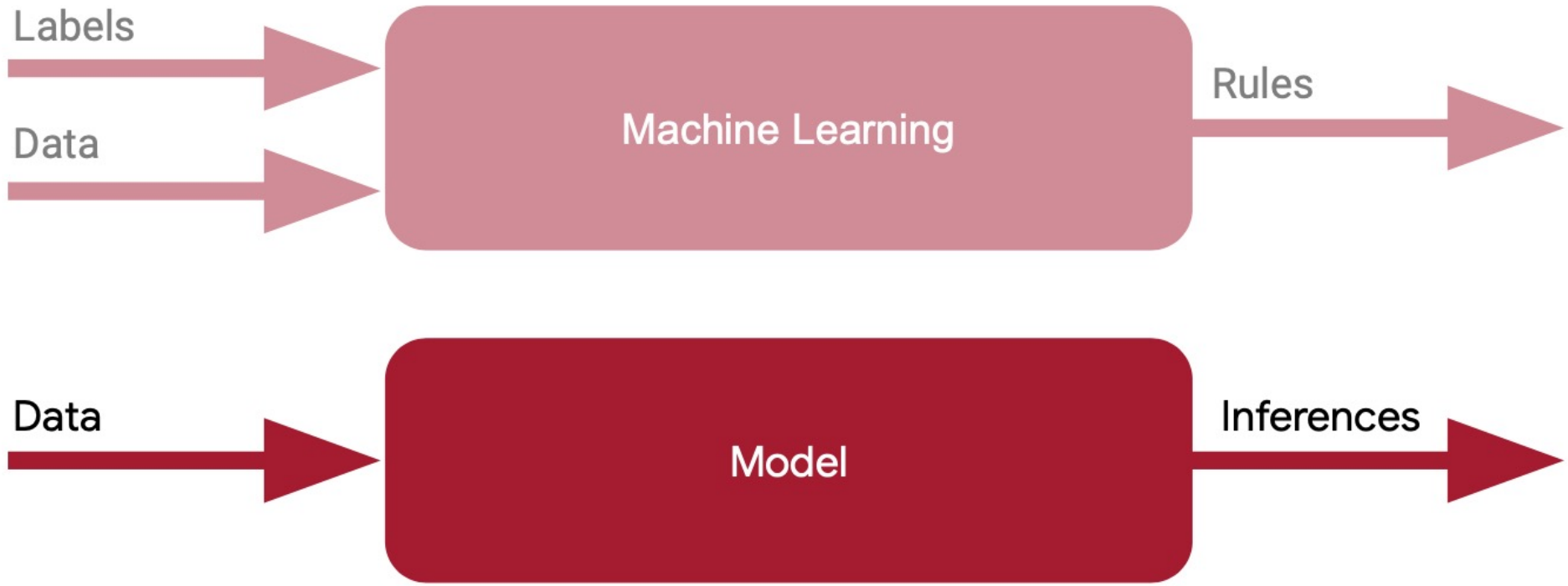
```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING

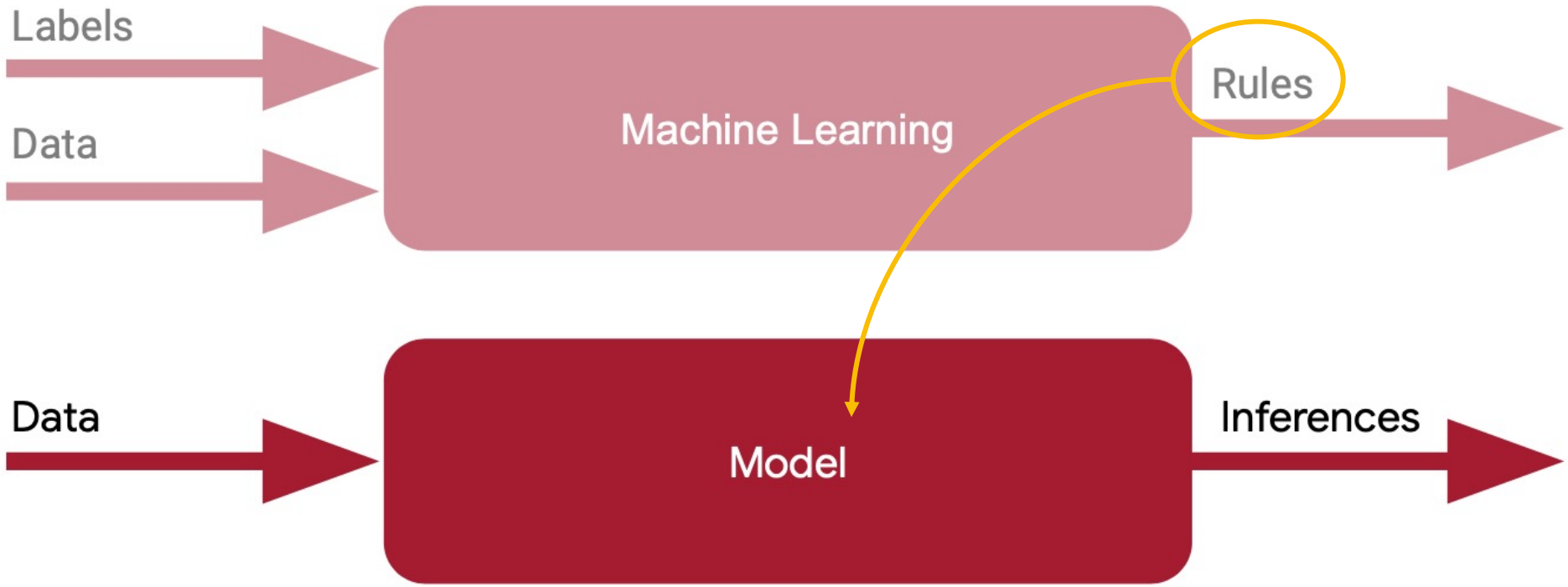
The Machine Learning Paradigm

The Machine Learning Paradigm

The Machine Learning Paradigm



The Machine Learning Paradigm



Thinking about loss…

A way to measure your accuracy

Matching X to Y

$X = \{ -1, 0, 1, 2, 3, 4 \}$

$Y = \{ -3, -1, 1, 3, 5, 7 \}$

Matching X to Y

$X = \{ -1, 0, 1, 2, 3, 4 \}$

$Y = \{ -3, -1, 1, 3, 5, 7 \}$

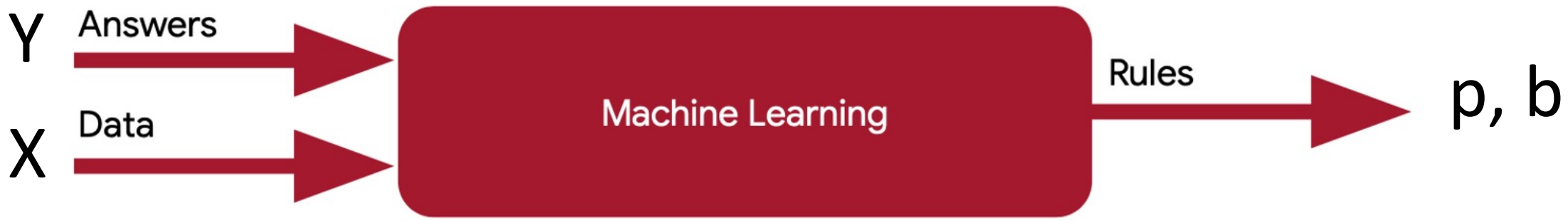
$$Y = p*X + b$$

Matching X to Y

X = { -1, 0, 1, 2, 3, 4 }

Y = { -3, -1, 1, 3, 5, 7 }

$$Y = p*X + b$$



Matching X to Y

Input

X = { -1, 0, 1, 2, 3, 4 }

Y = { -3, -1, 1, 3, 5, 7 }

Target

Y = p*X + b

Parameters

Make a guess!

Y = 3X - 1

p = +3
b = -1

X = { -1, 0, 1, 2, 3, 4 }

Y = { -4, -1, 2, 5, 8, 11 }

How good is the guess?

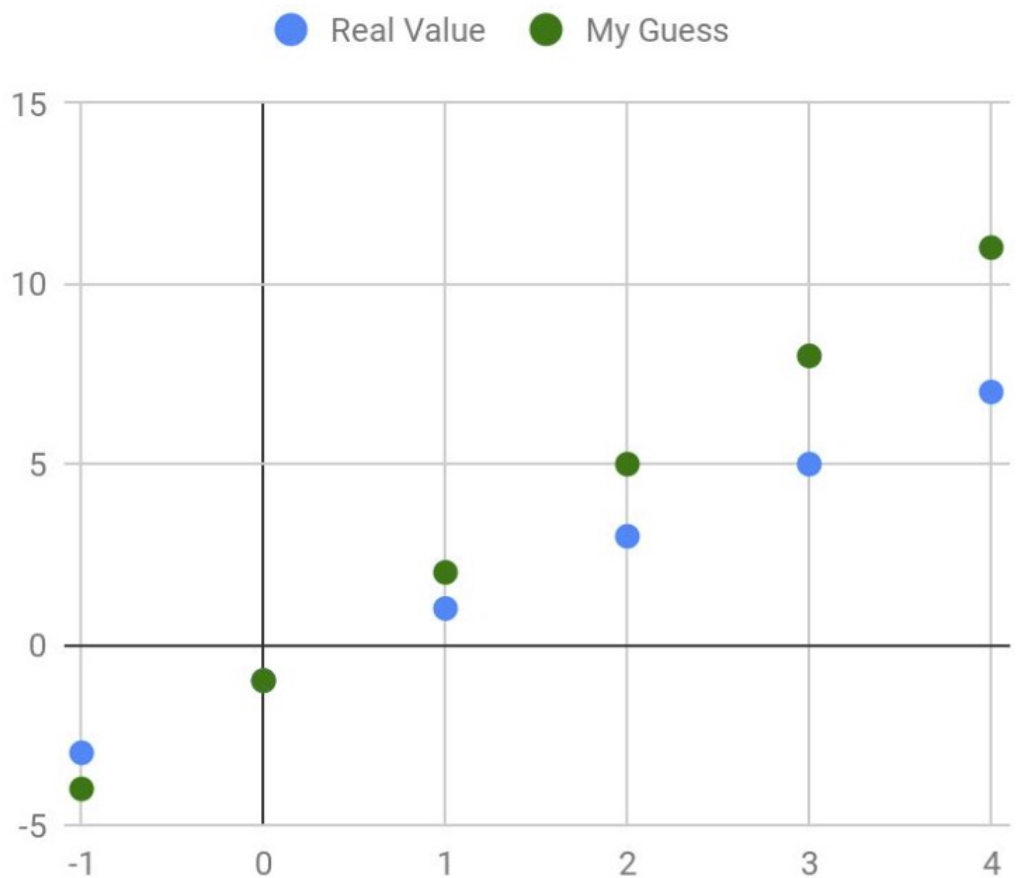
Y = 3X - 1

X = { -1, 0, 1, 2, 3, 4 }

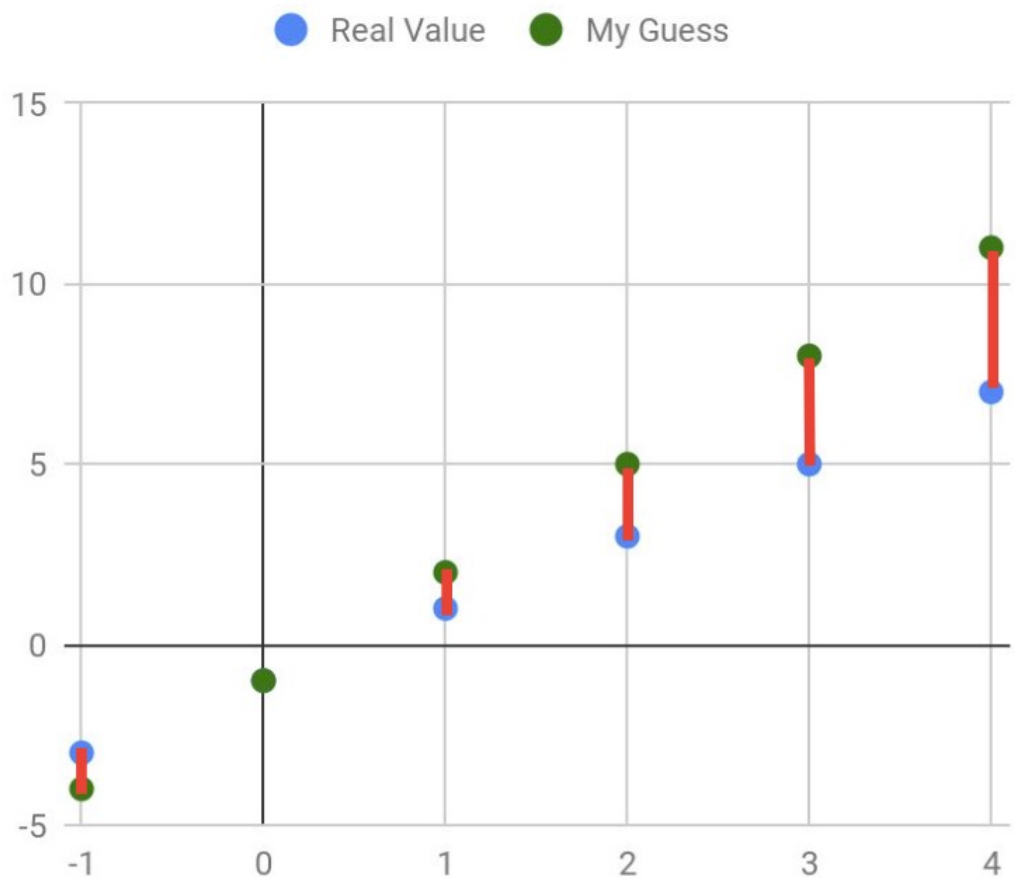
My Y = { -4, -1, 2, 5, 8, 11 }

Real Y = { -3, -1, 1, 3, 5, 7 }

Let’s measure it!

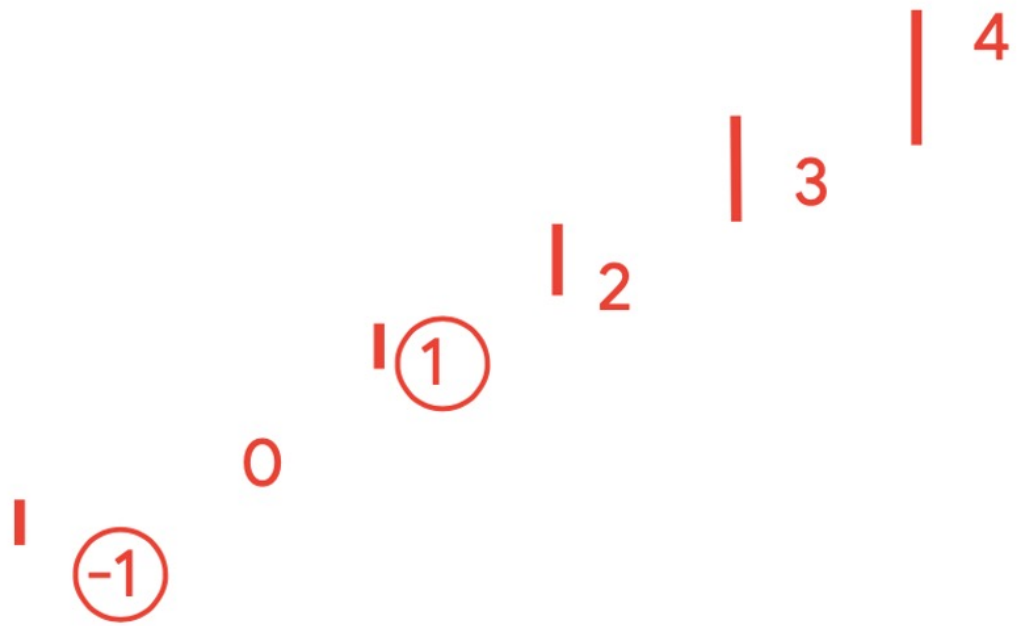


Let’s measure it!



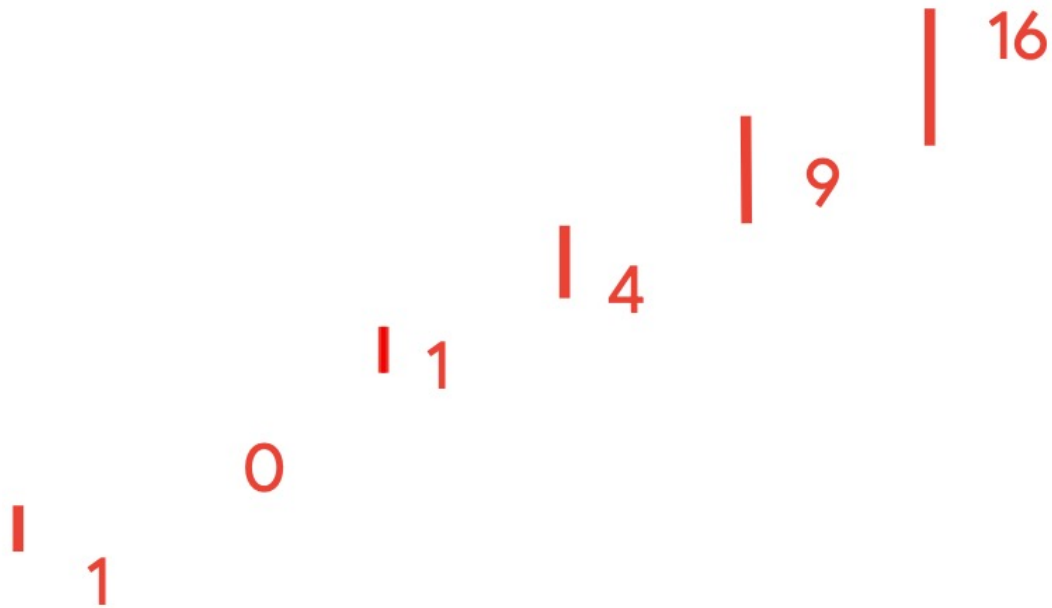
Let’s measure it!

Houston, we have a problem!



Houston, we have a problem!

What if we **square**^{2} them?



Total that (Σ) and take the square root √

sqrt(1 + 1 + 4 + 9 + 16)

= sqrt(31)
= 5.57

Make another guess!

Y = 2X - 2

p = +2
b = -2

X = { -1, 0, 1, 2, 3, 4 }

My Y = { -4, -2, 0, 2, 4, 6 }

Real Y = {-3, -1, 1, 3, 5, 7}

Diff^2 = {1, 1, 1, 1, 1}

Get the same difference, repeat the same process.

sqrt(1 + 1 + 1 + 1 + 1)

= sqrt(5)
= 2.23

Make another guess!

Y = 2X - 1

X = { -1, 0, 1, 2, 3, 4 }

My Y = { -3, -1, 1, 3, 5, 7 }

Real Y = {-3, -1, 1, 3, 5, 7}

$Diff^2$ = {0, 0, 0, 0, 0}

Make another guess!

Y = 2X - 1

Parameters

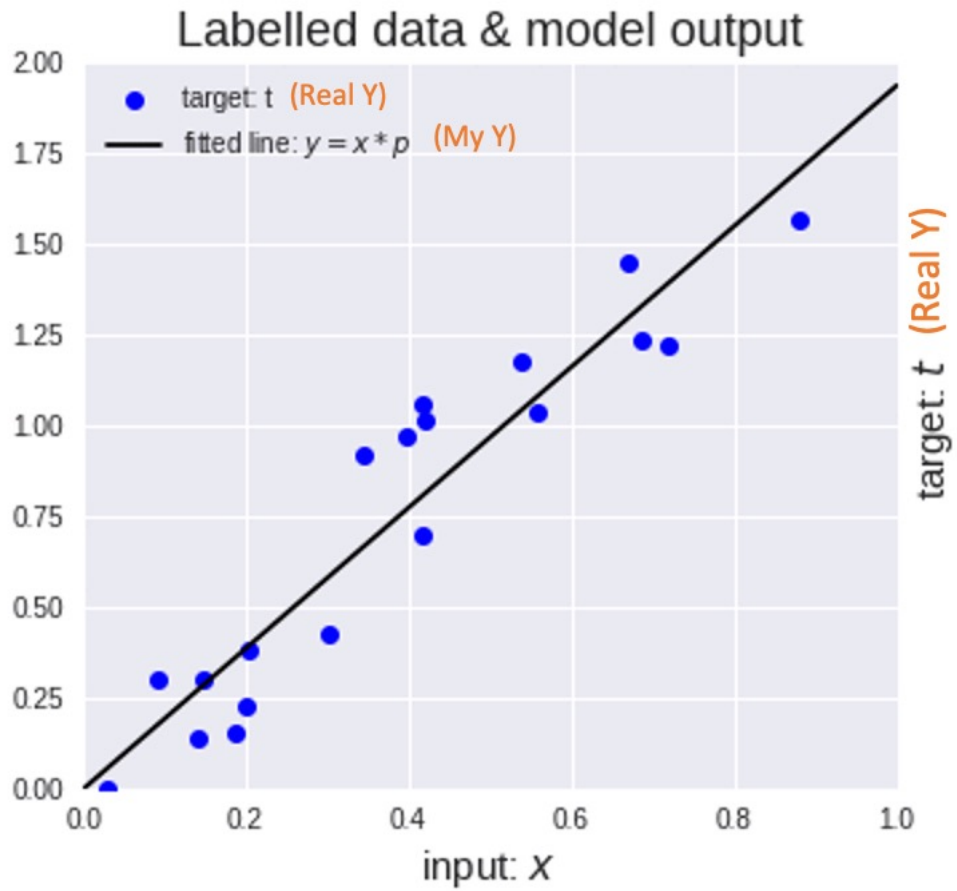
X = { -1, 0, 1, 2, 3, 4 }

My Y = { -3, -1, 1, 3, 5, 7 }

Real Y = {-3, -1, 1, 3, 5, 7}

Diff^2 = {0, 0, 0, 0, 0}

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$



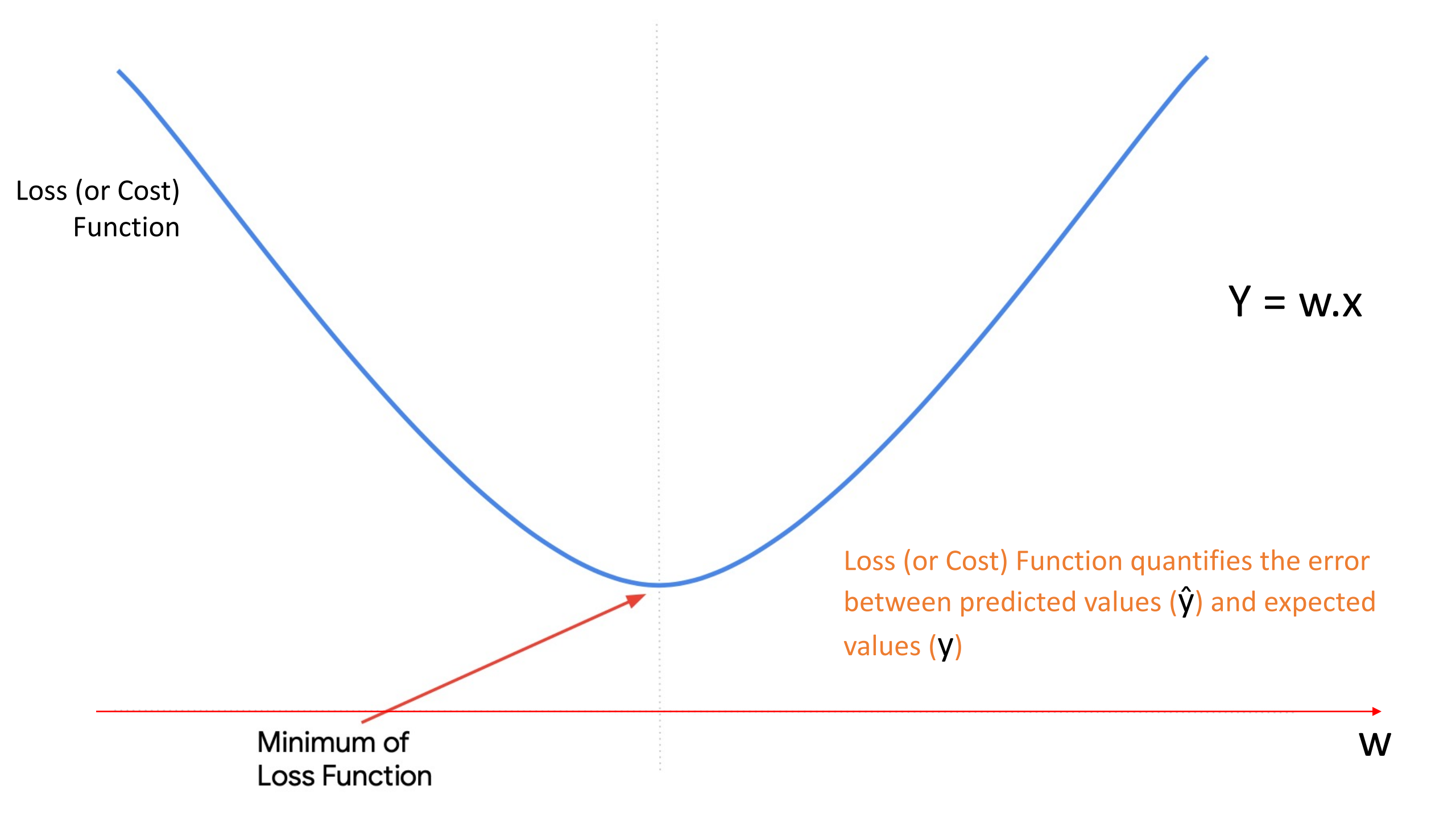
Mean Squared Error: Goal: Minimum as possible!

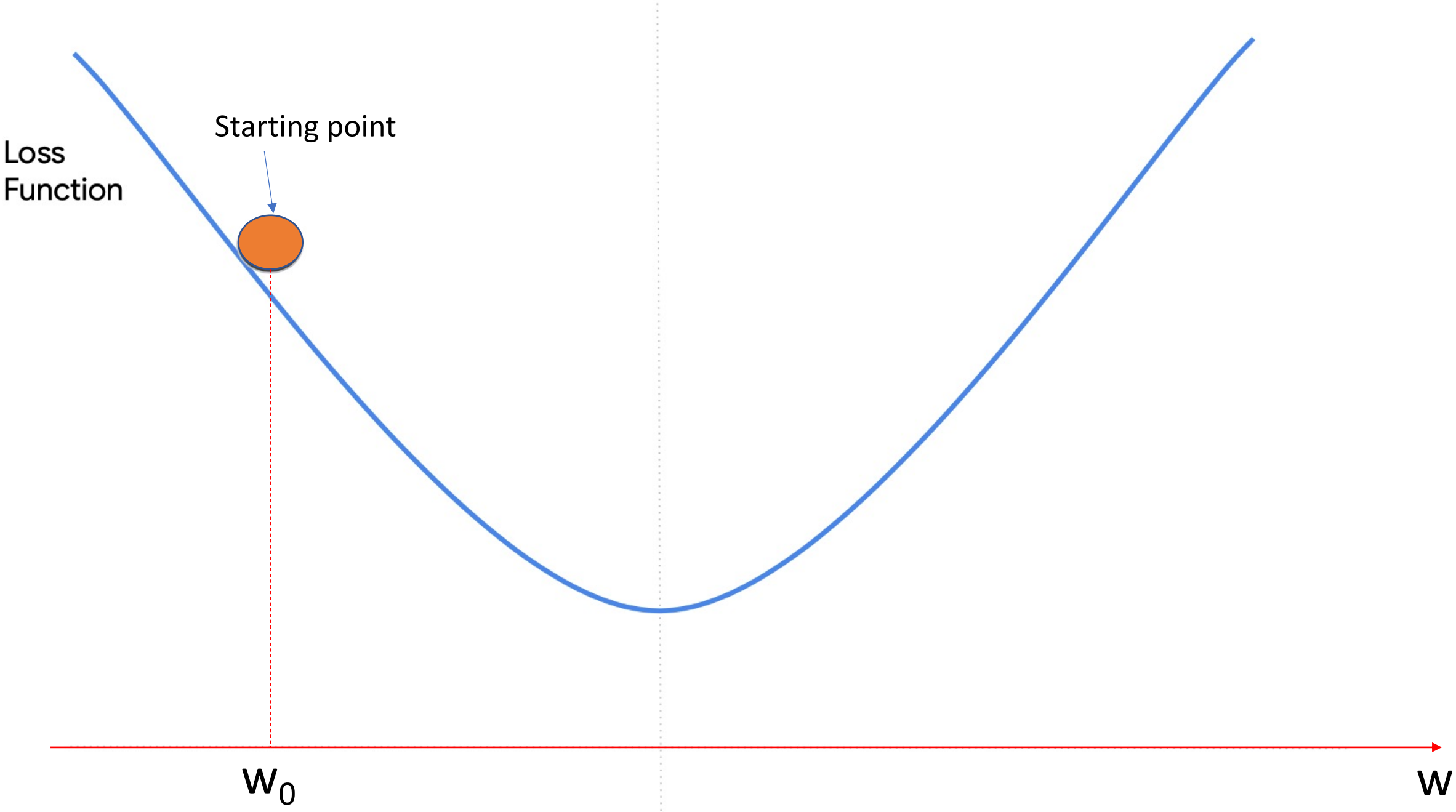
Exploring Loss and Cost Function

Code Time!

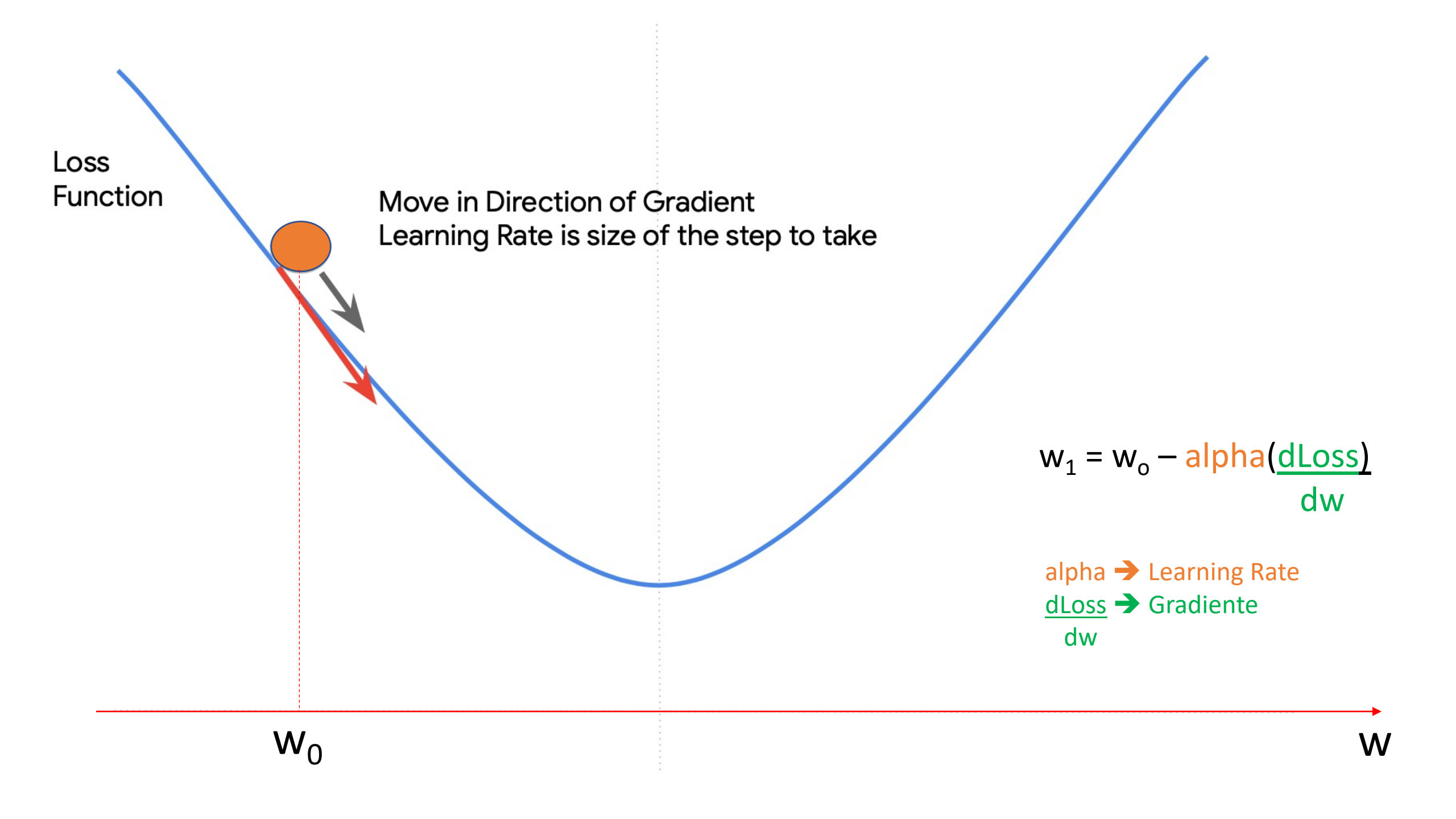
Minimizing loss…

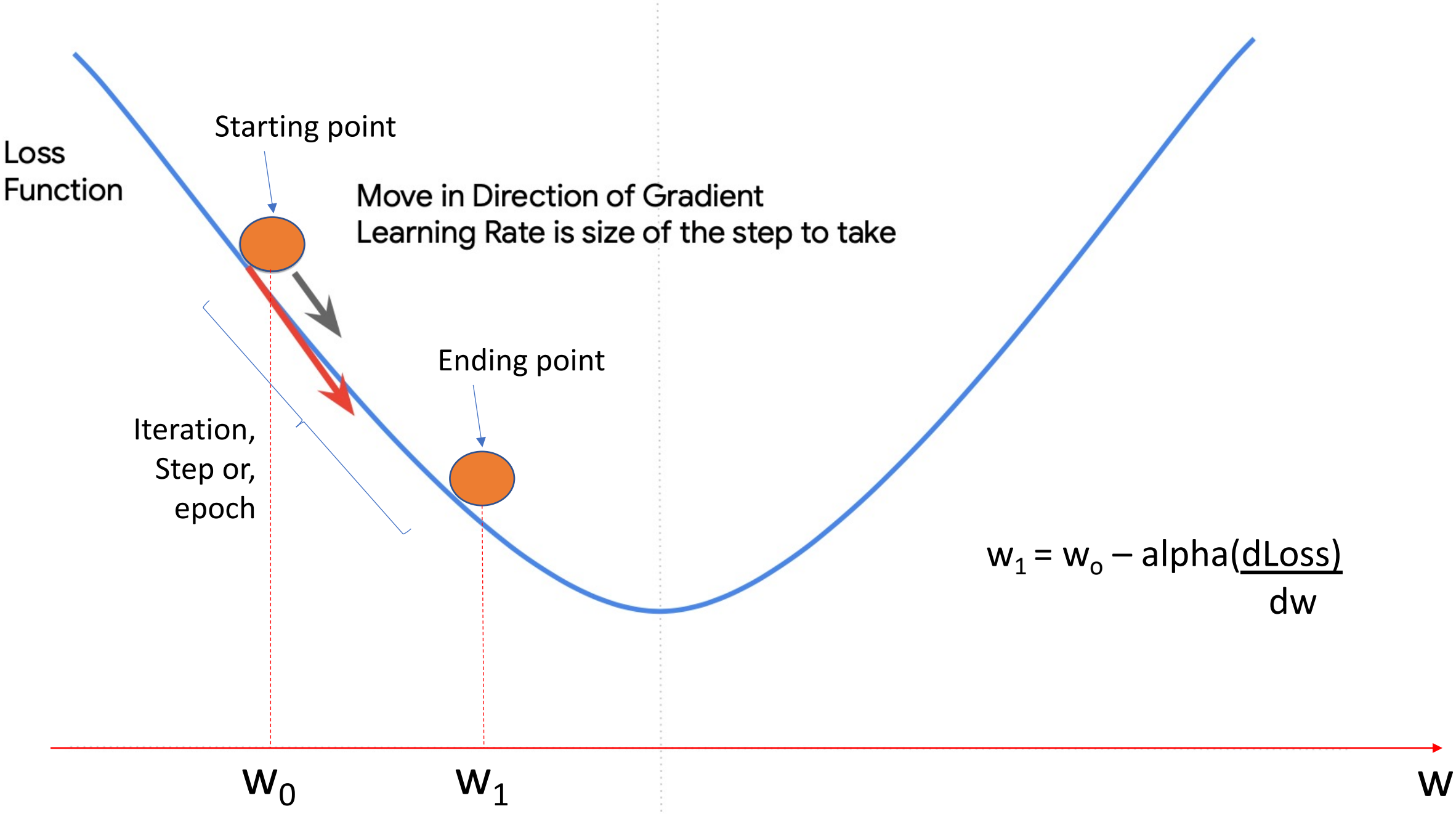
Moving down the curve…

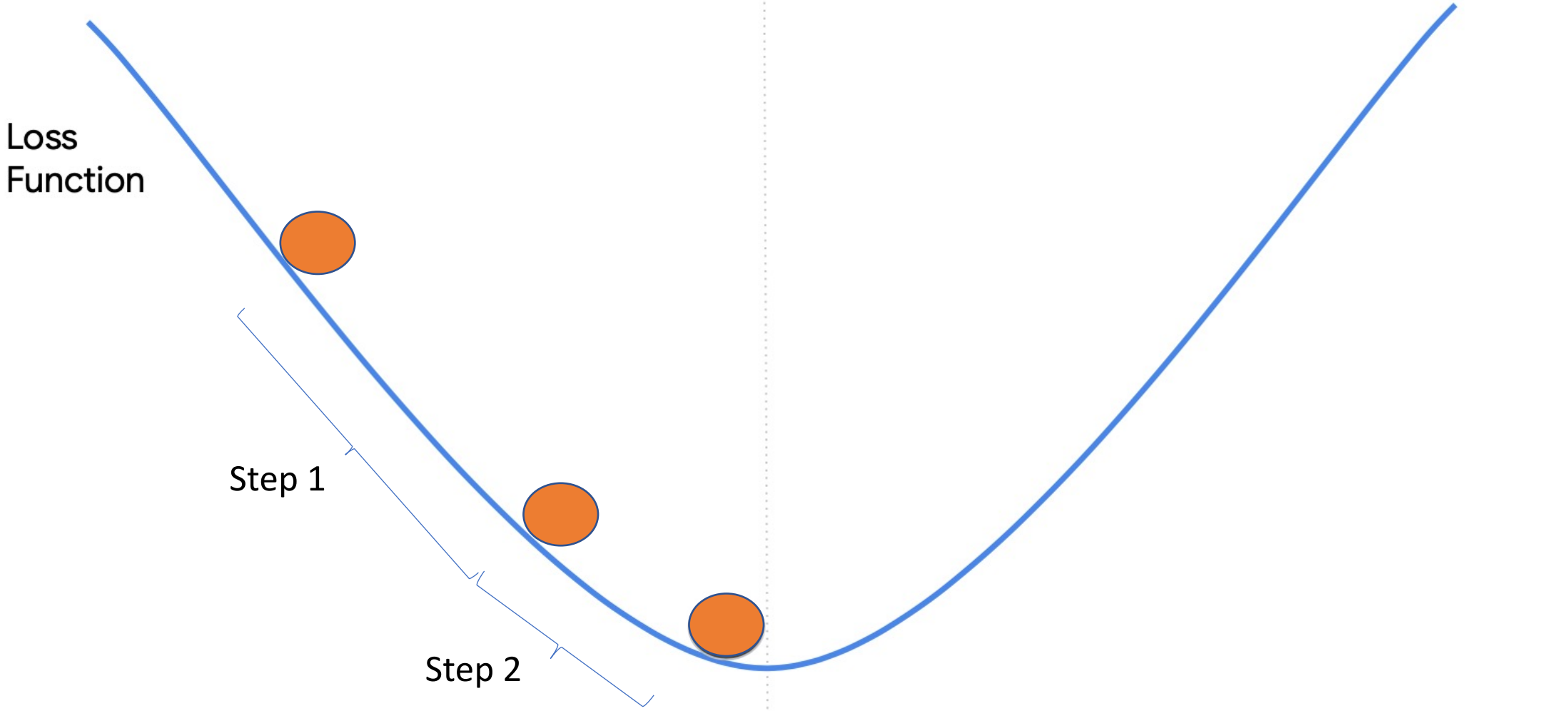

Loss (or Cost)
Function
Y = w.x
Loss (or Cost) Function quantifies the error between predicted values (ŷ) and expected values (y)
Minimum of
Loss Function
w


Loss
Function
Starting point
w_0
w


Loss
Function
Gradient of
value
Goal: To minime our loss function by moving in
a direction indicated of the gradient
w_0
w

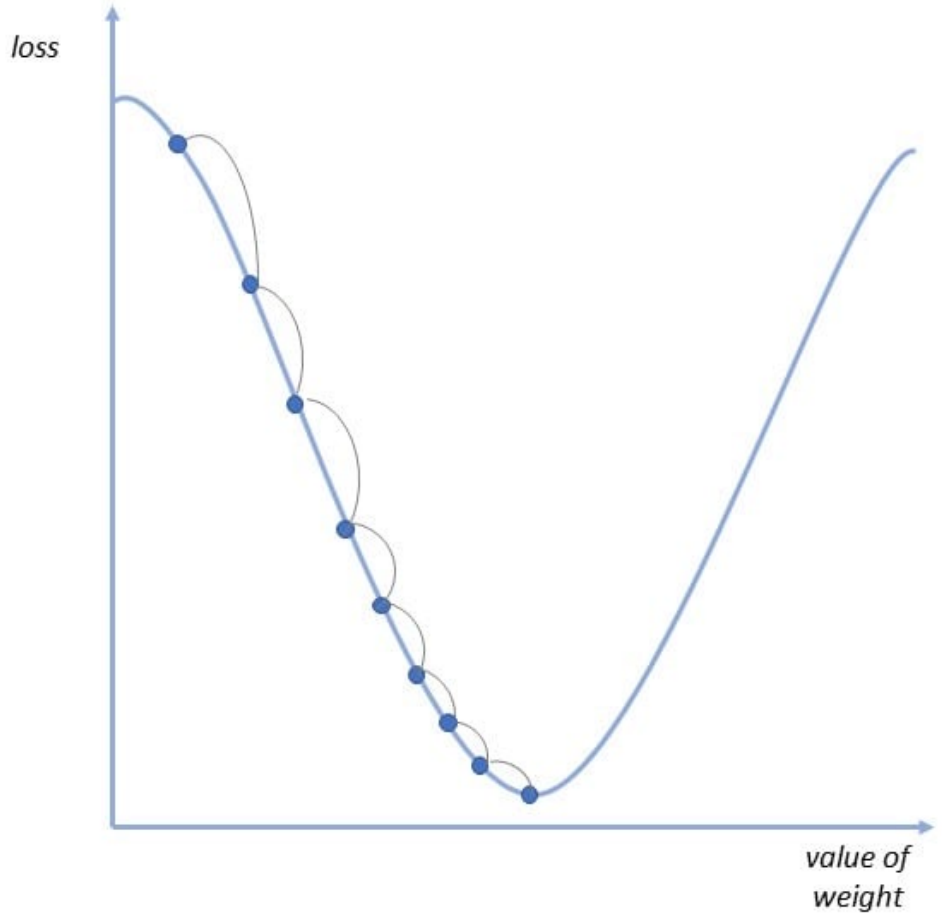

Loss
Function
Move in Direction of Gradient
Learning Rate is size of the step to take
$w_1 = w_o - \text{alpha}\left(\frac{dLoss}{dw}\right)$
alpha ➔ Learning Rate
$\frac{dLoss}{dw}$ ➔ Gradiente
w_0
w


Loss Function
Starting point
Move in Direction of Gradient
Learning Rate is size of the step to take
Ending point
Iteration, Step or, epoch
$w_1 = w_o - \text{alpha}(\frac{dLoss}{dw})$
w_0
w_1
w

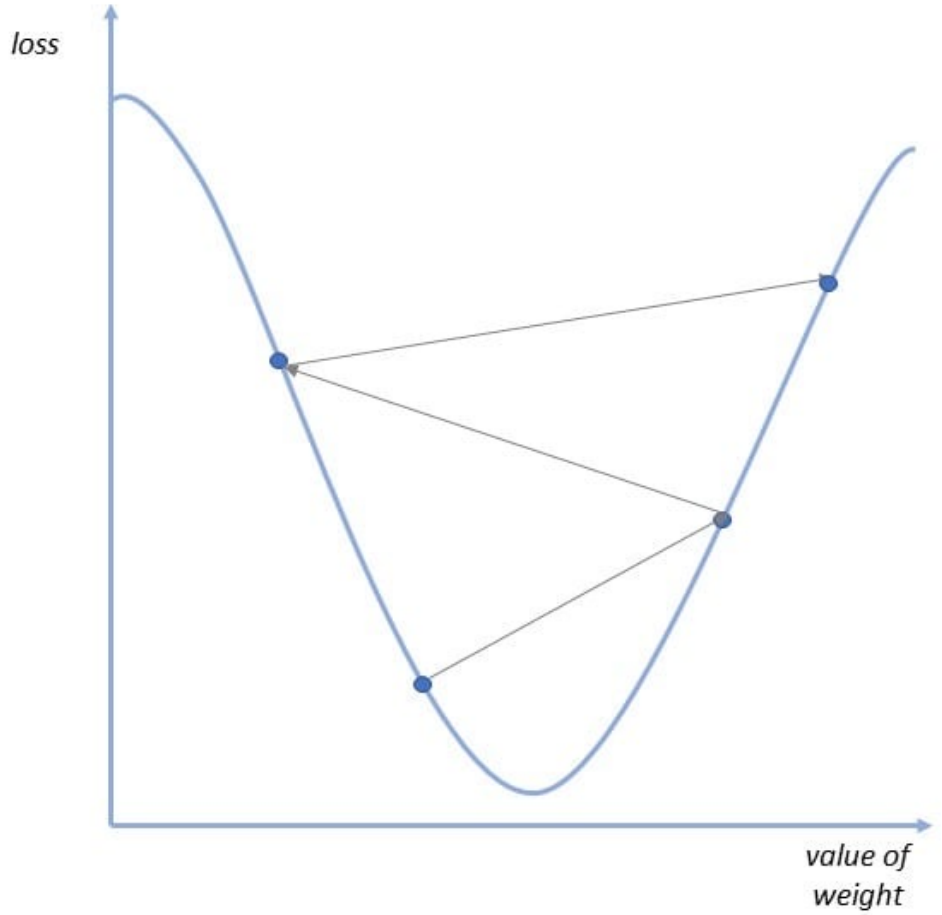

Loss
Function
Step 1
Step 2

It is important to choose the correct Learning Rate (size of the step)

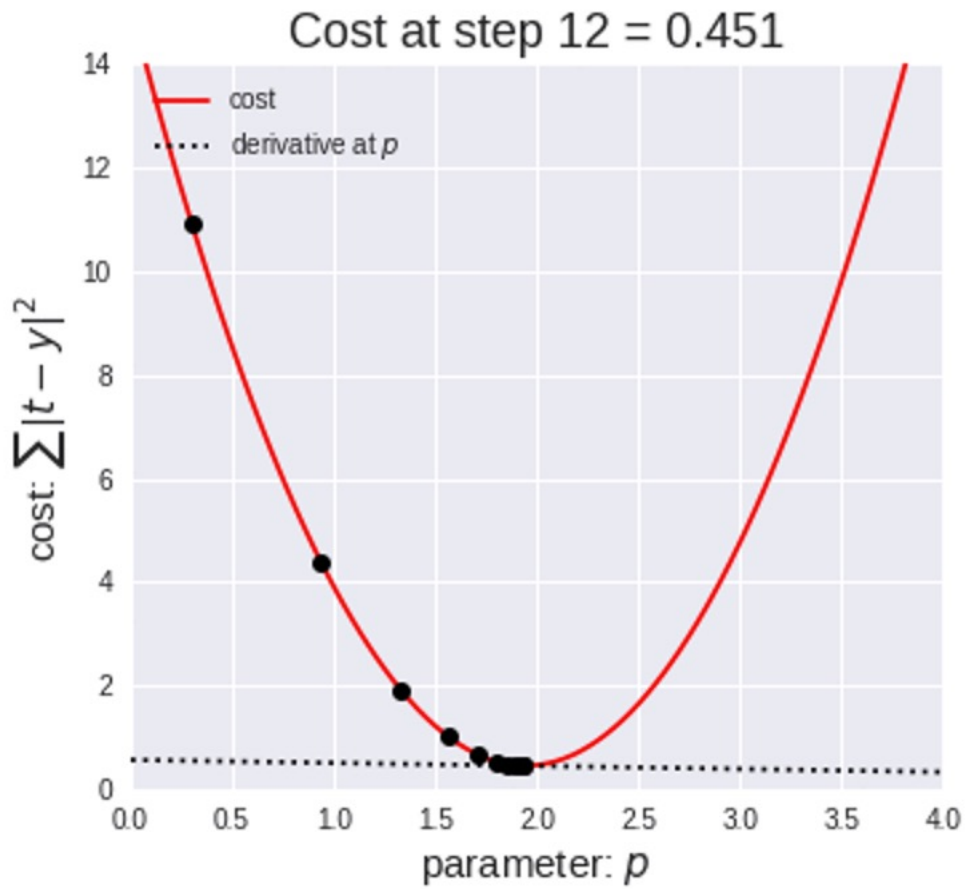
If the **Learning Rate** is too small it may take a long time to reach the minimum

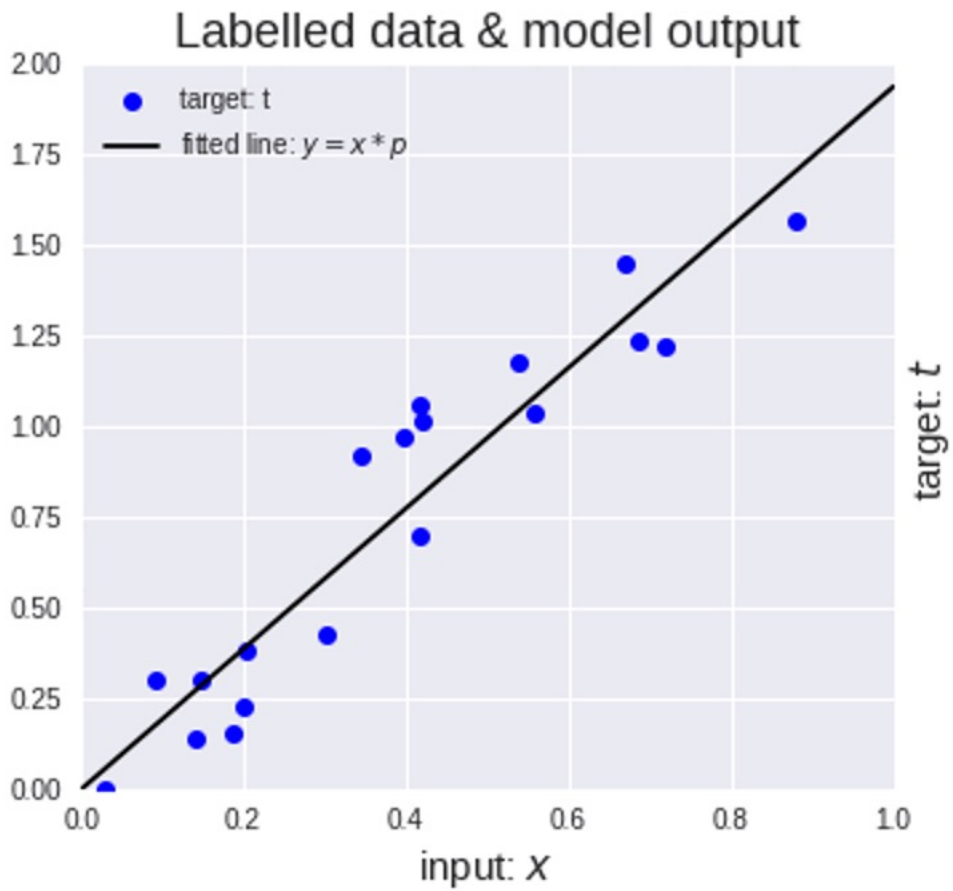


If the **Learning Rate** is too large we may never reach the minimum



Gradient Descent algorithm





The Machine Learning Paradigm

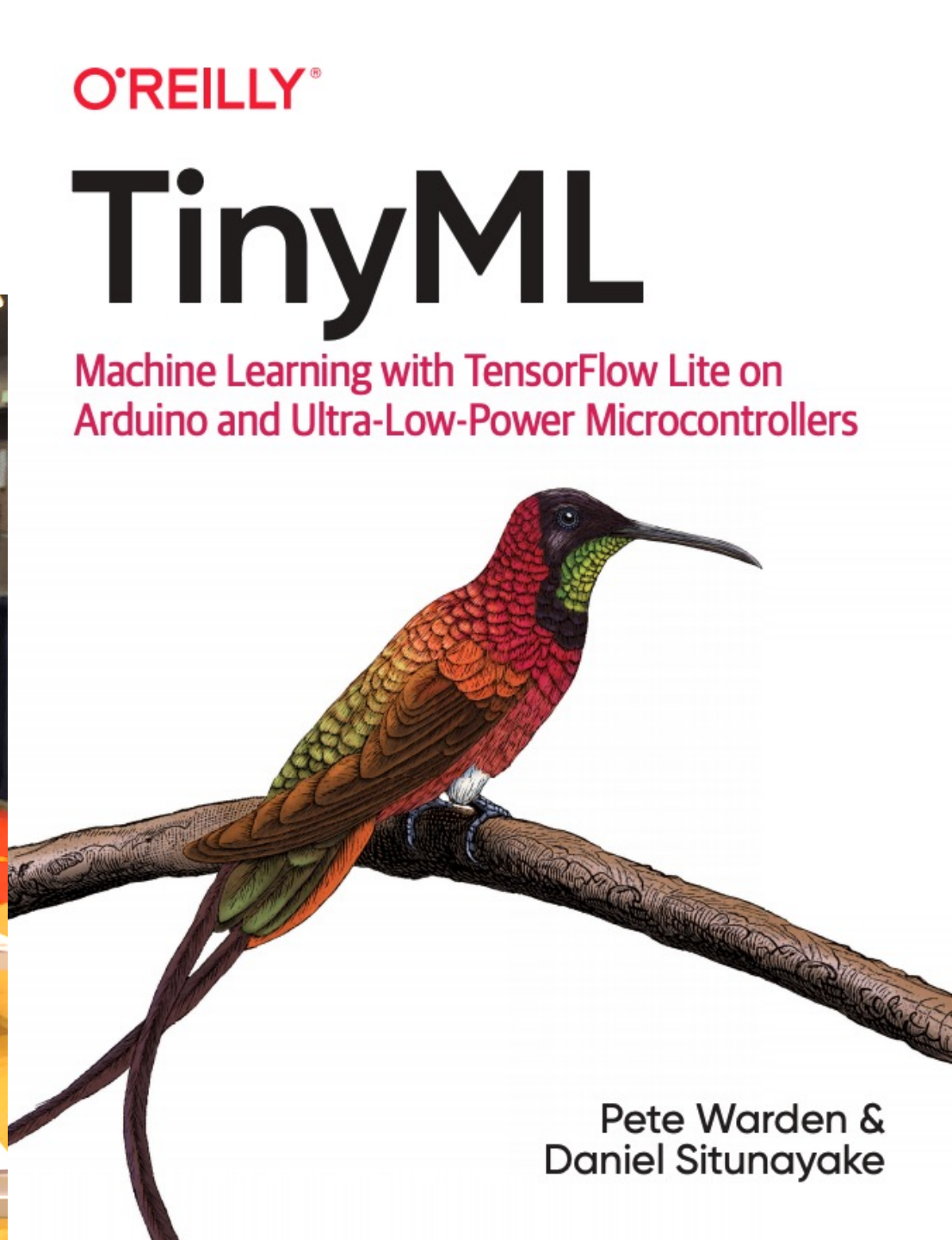
The Machine Learning Paradigm

Reading Material

Special IESTI01 Class with **Daniel Situnayake**,

ML R&D Lead at Edge Impulse, guest Lecturer at Harvard and Berkeley Universities. Ex-GoogleAI and Co-author of TinyML.

Reserve in your calendar: July 14th at 4:30pm (Brasilia Time)



Main references

- Harvard School of Engineering and Applied Sciences - CS249r: Tiny Machine Learning
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
- Introduction to Embedded Machine Learning (Coursera)
- Text Book: "TinyML" by Pete Warden, Daniel Situnayake

I want to thank Laurence Moroney from Google, Harvard professor Vijay Janapa Reddi, Ph.D. student Brian Plancher and their staff for preparing the excellent material on TinyML that is the basis of this course at UNIFEI.

The IESTI01 course is part of the TinyML4D, an initiative to make TinyML education available to everyone globally.

Thanks

And stay safe!