EECE-4710 “IoT and TinyML”

Machine Learning (ML) Paradigm

Cris Ababei

MARQUETTE UNIVERSITY

BE THE DIFFERENCE.

1

1

ML Paradigm: From Coding to Learning

2

2

Traditional Coding or Programming Paradigm

- Rules that determine program behavior are well defined
- Programmer pre-calculates and pre-determines everything
- Scenarios limited by program complexity

3

3

Example: Activity Detection

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

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

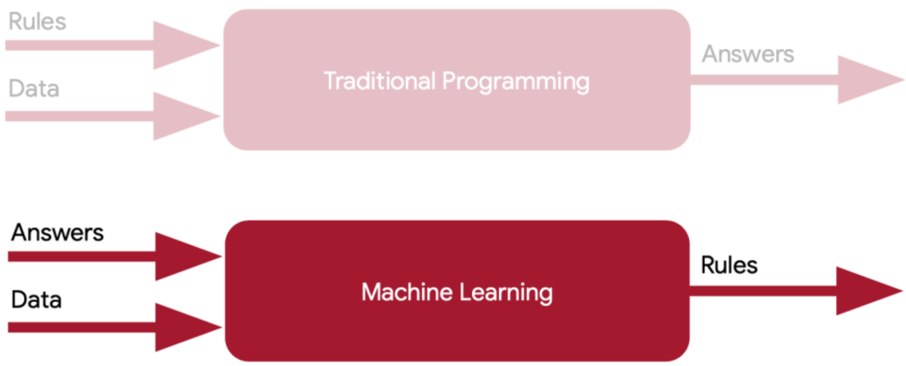
```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// ???
```

4

4

The Traditional Programming Paradigm



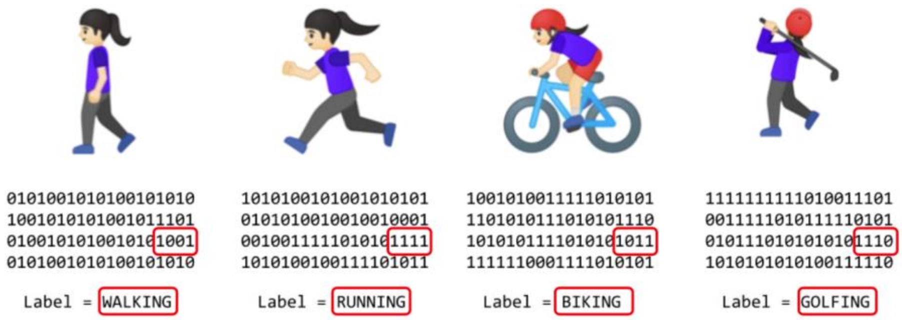
5

Activity Detection with Machine Learning

0101001010100101010	1010100101001010101	1001010011111010101	1111111111010011101
1001010101001011101	0101010010010010001	1101010111010101110	0011111010111110101
0100101010010101001	0010011111010101111	1010101111010101011	0101110101010101110
0101001010100101010	1010100100111101011	1111110001111010101	1010101010100111110
Label = WALKING	Label = RUNNING	Label = BIKING	Label = GOLFING

6

The Machine Learning Paradigm



7

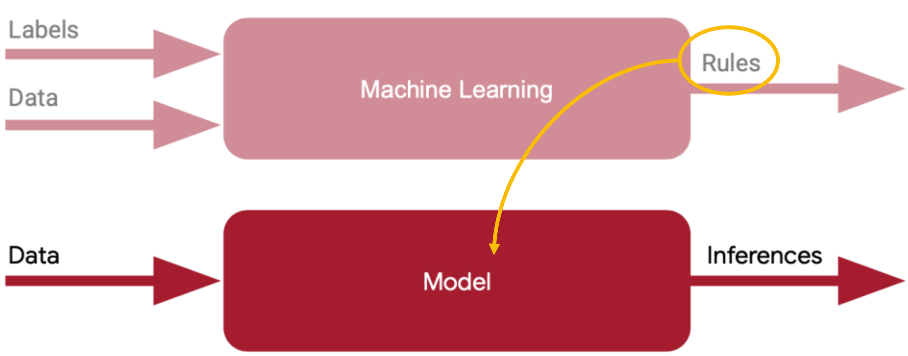
The Machine Learning Paradigm

8

The Machine Learning Paradigm

9

The Machine Learning Paradigm

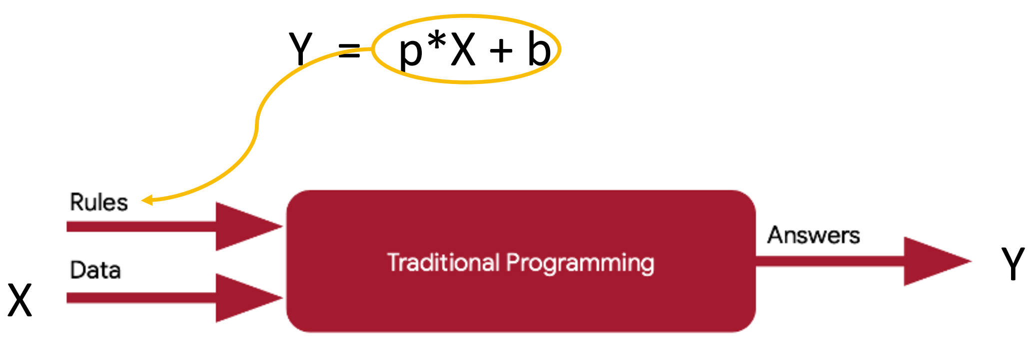


10

Loss:
A Way to Measure Accuracy

11

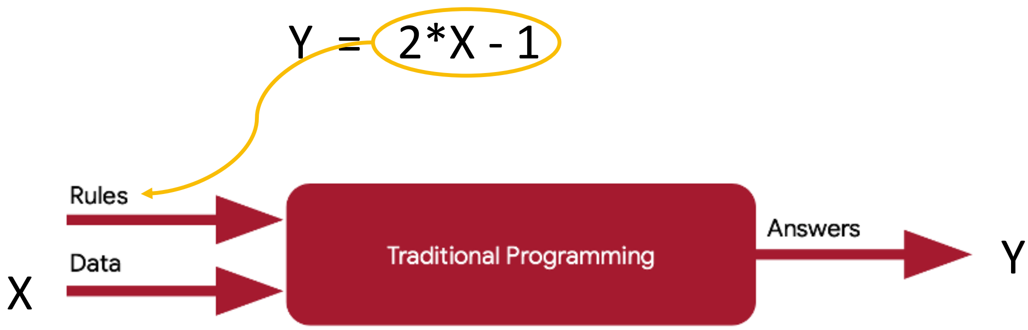
11



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

Y = {?, ?, ? ,?, ?, ?}

12



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

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

13

Matching X to Y

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

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

14

Matching X to Y

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

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

$Y = p*X + b$

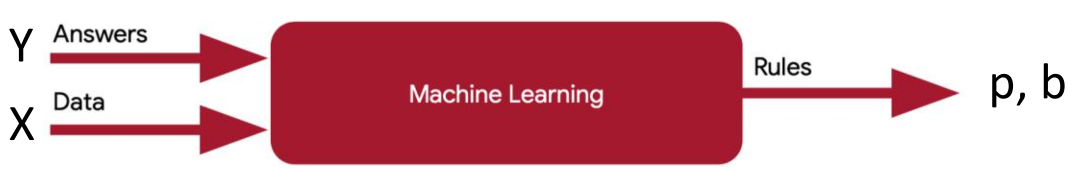
15

Matching X to Y

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

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

$Y = p*X + b$



16

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

17

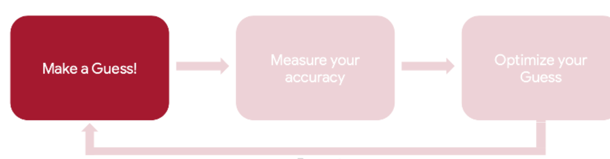
Make a guess!

$Y = 3X - 1$

Parameters

p = +3

b = -1



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

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

18

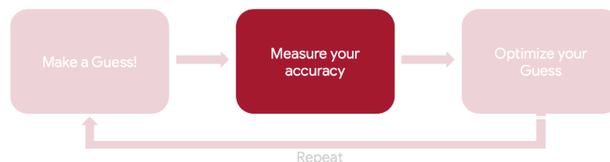
How good is the guess?

Y = 3X - 1

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

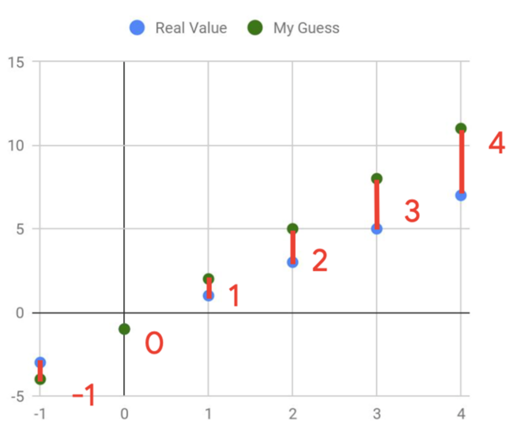
My Y = { -4, -1, 2, 5, 8, 11 }

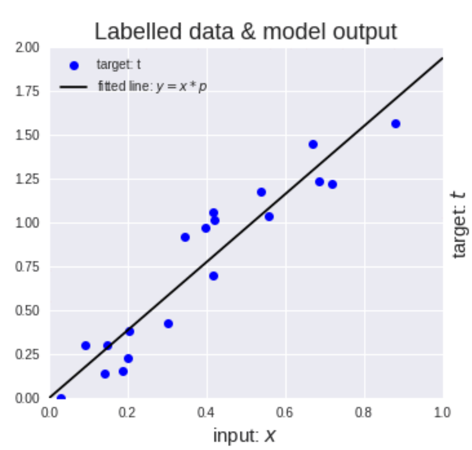
Real Y = { -3, -1, 1, 3, 5, 7 }



19

Let’s measure it

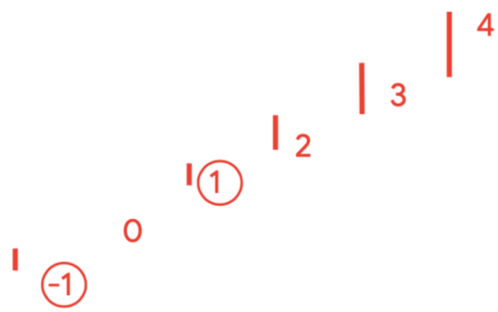




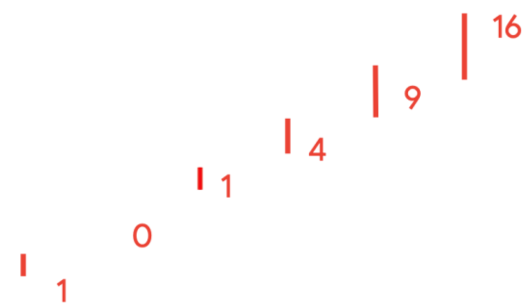
20

20

Houston, we have a problem!



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

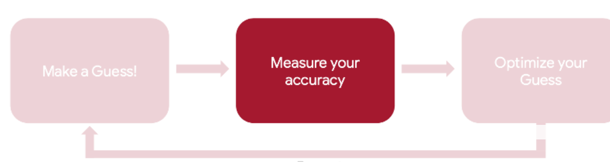


21

Calculate de mean squared error:

= (1 + 1 + 4 + 9 + 16) / 6

= 5.17



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

Mean Squared Error

Goal: Minimum as possible!

22

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 }

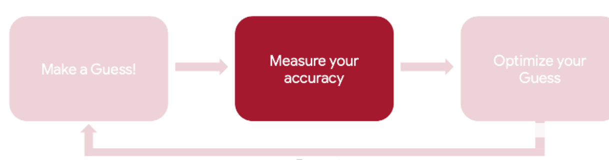
Real Y = {-3, -1, 1, 3, 5, 7}

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

23

Get the same difference, repeat the same process.

= (1 + 1 + 1 + 1 + 1 + 1) / 6
= 1.00



24

Make another guess!

Y = 2X - 1

p = +2
b = -1

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

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

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

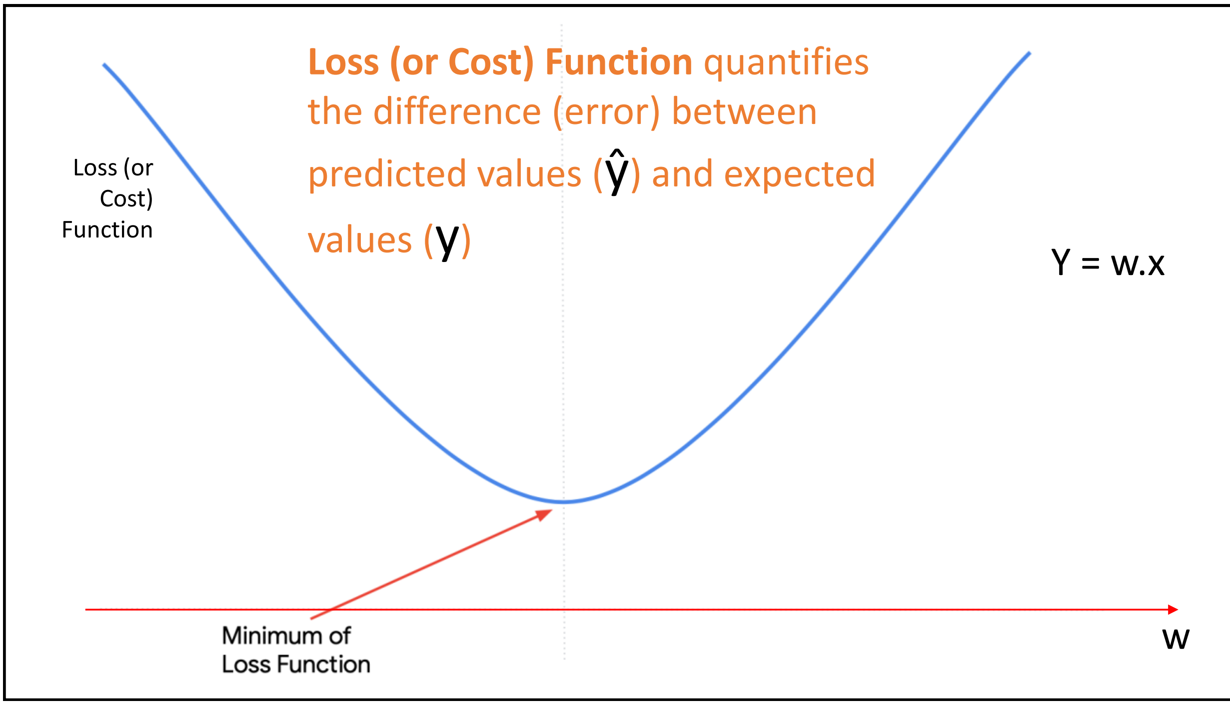
Diff^2 = {0, 0, 0, 0, 0, 0}

25

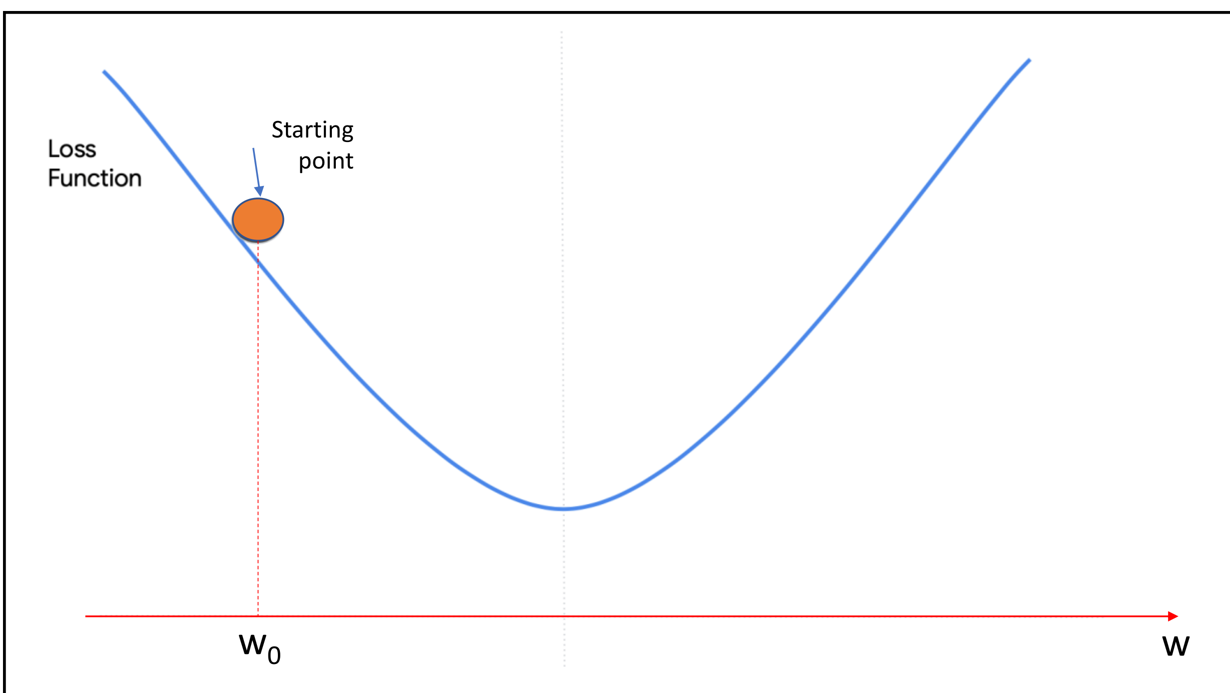
Minimize Loss

26

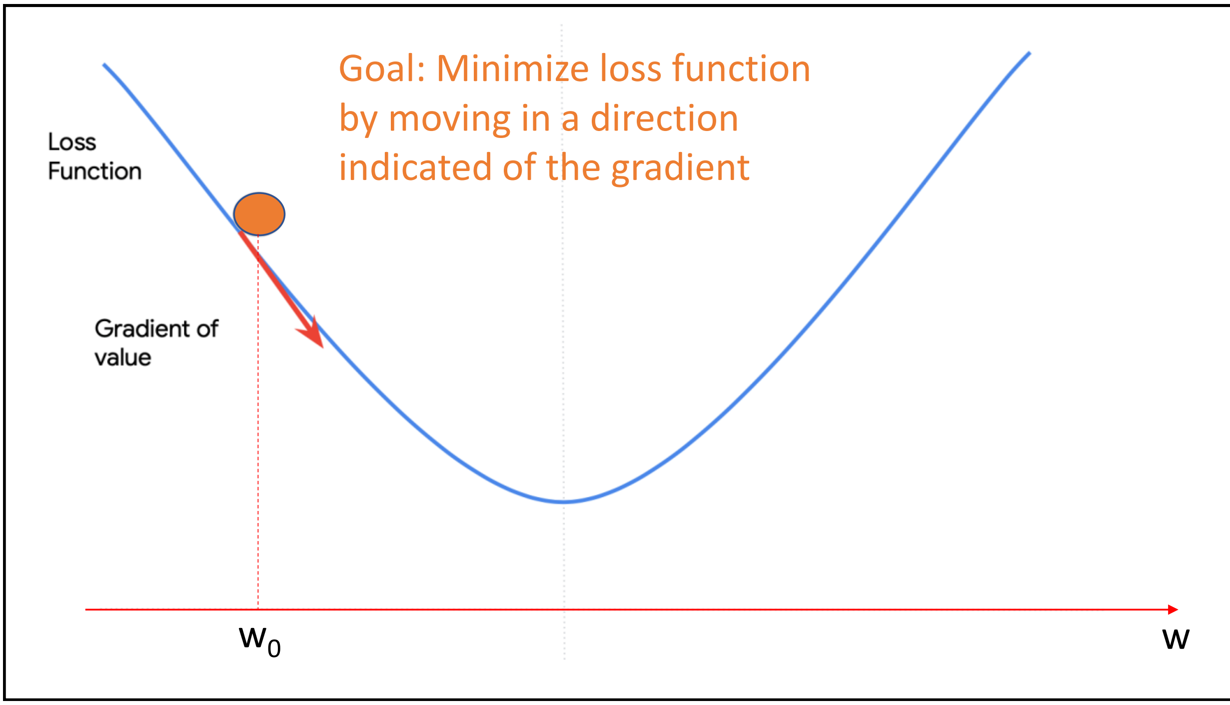
26


Loss (or Cost) Function quantifies the difference (error) between predicted values ($\hat{y}$) and expected values (y)
Loss (or Cost) Function
Y = w.x
Minimum of Loss Function
w

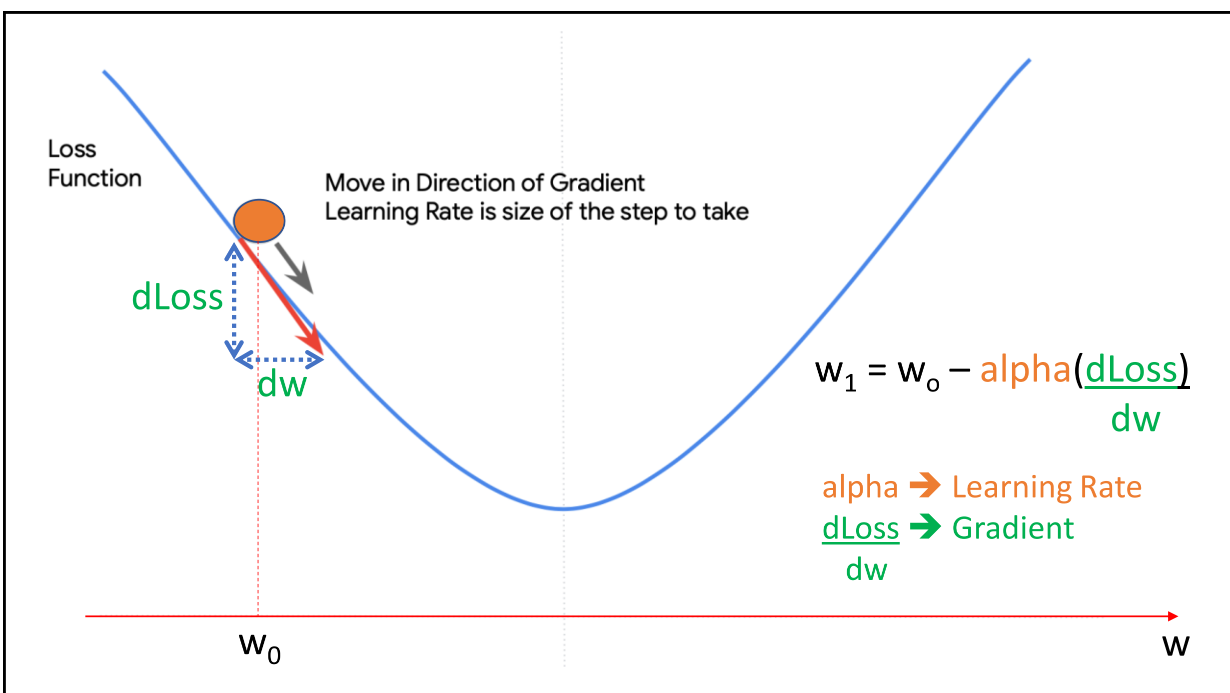
27


Loss Function
Starting point
w_0
w

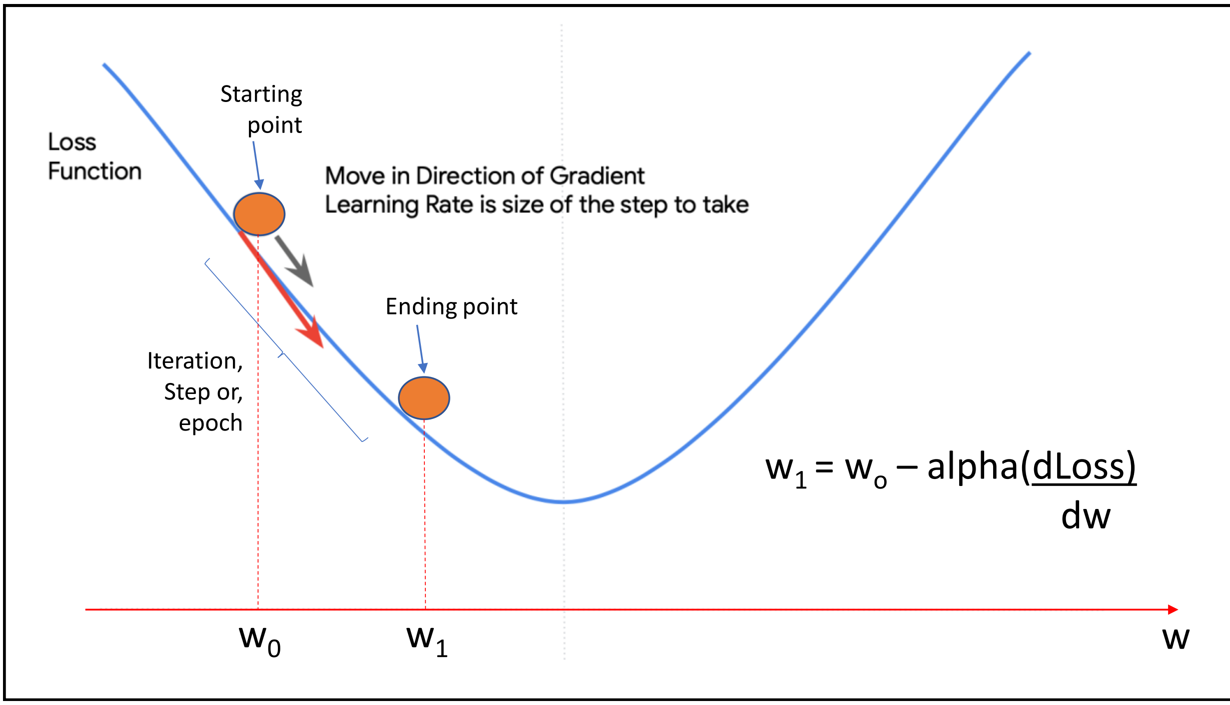
28


Goal: Minimize loss function
by moving in a direction
indicated of the gradient
Loss
Function
Gradient of
value
w0
w

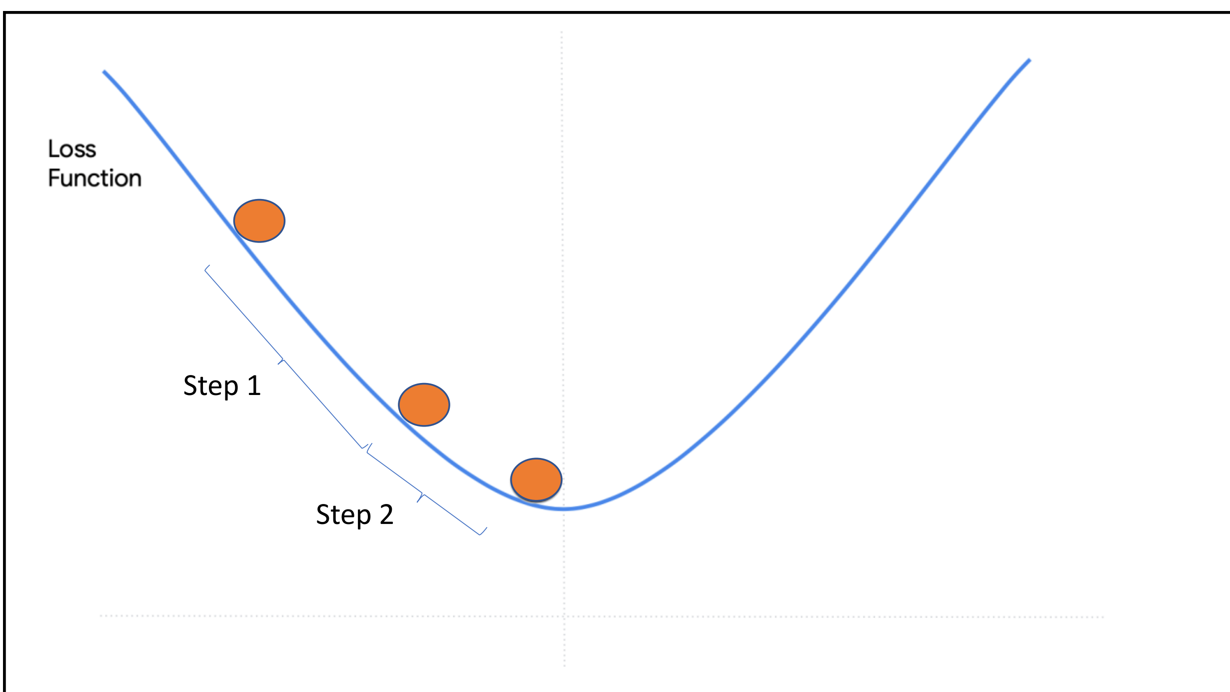
29


Loss
Function
Move in Direction of Gradient
Learning Rate is size of the step to take
dLoss
dw
w1 = wo – alpha(dLoss)
dw
alpha → Learning Rate
dLoss → Gradient
dw
w0
w

30


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

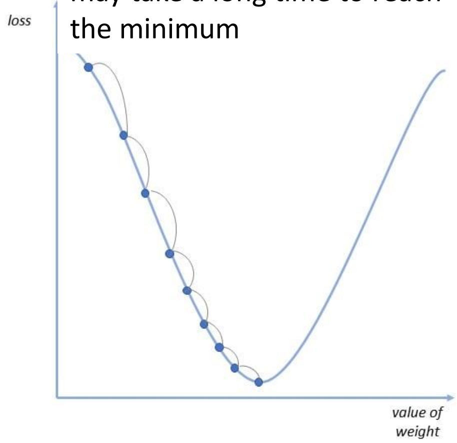
31


Loss
Function
Step 1
Step 2

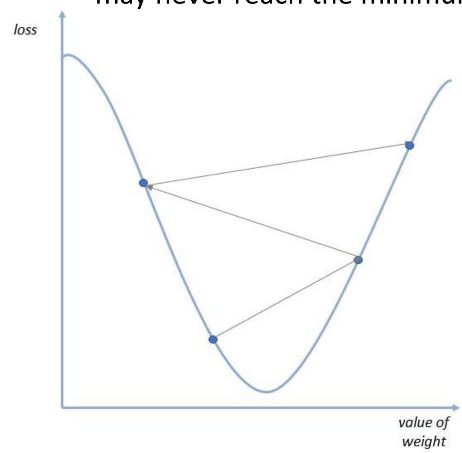
32

Important to Choose Correct Learning Rate (size of the step)

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

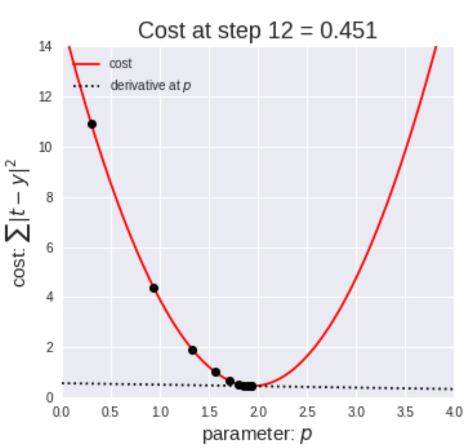


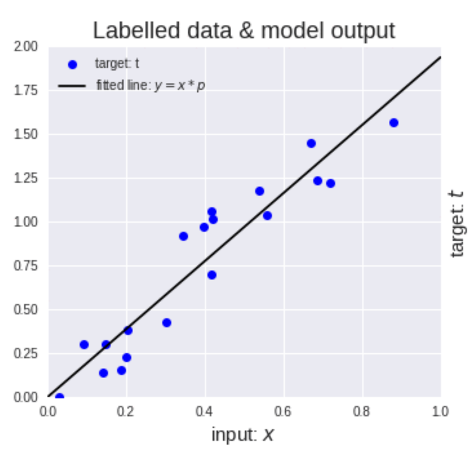
If **Learning Rate** is too large, it may never reach the minimum



33

Gradient Descent Algorithm





34

The Machine Learning Paradigm

W_i , b_i
Make a Guess!
(Weights, Biases)

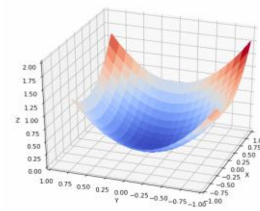
→

MSE
Measure your accuracy
(Loss, Accuracy)

→

SGD
Optimize your Guess
(Learning Rate)

Epochs
(Back-Propagation)



35

Credits

- A previous edition of this course was developed in collaboration with Dr. Susan C. Schneider of Marquette University.
- We are very grateful and thank all the following professors, researchers, and practitioners for jump-starting courses on TinyML and for sharing their teaching materials:
- Prof. Marcelo Rovai - TinyML - Machine Learning for Embedding Devices, UNIFEI
  - https://github.com/Mjrovai/UNIFEI-IESTI01-TinyML-2022.1
- Prof. Vijay Janapa Reddi - CS249r: Tiny Machine Learning, Applied Machine Learning on Embedded IoT Devices, Harvard
  - https://sites.google.com/g.harvard.edu/tinyml/home
- Prof. Rahul Mangharam – ESE3600: Tiny Machine Learning, Univ. of Pennsylvania
  - https://tinyml.seas.upenn.edu/#
- Prof. Brian Plancher - Harvard CS249r: Tiny Machine Learning (TinyML), Barnard College, Columbia University
  - https://a2r-lab.org/courses/cs249r_tinyml/

36

36

References

- Additional references from where information and other teaching materials were gathered include:
- Applications & Deploy textbook: “TinyML” by Pete Warden, Daniel Situnayake
  - https://www.oreilly.com/library/view/tinyml/9781492052036/
- Deploy textbook “TinyML Cookbook” by Gian Marco Iodice
  - https://github.com/PacktPublishing/TinyML-Cookbook
- Jason Brownlee
  - https://machinelearningmastery.com/
- TinyMLedu
  - https://tinyml.seas.harvard.edu/
- Professional Certificate in Tiny Machine Learning (TinyML) – edX/Harvard
  - https://www.edx.org/professional-certificate/harvardx-tiny-machine-learning
- Introduction to Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/introduction-to-embedded-machine-learning
- Computer Vision with Embedded Machine Learning - Coursera/Edge Impulse
  - https://www.coursera.org/learn/computer-vision-with-embedded-machine-learning