

RADONC AI CURRICULUM

Introduction to Deep Learning

LECTURE 1: Background & Linear Regression

ANDREW Y. K. FOONG, PH.D.

March 26th 2026



Radiation
Oncology
AI & Data Analytics
AIDA

Today's lecture

1. What is AI?
 - *A bird's eye view of the field*
2. Goals and philosophy
 - *What is this course about?*
3. Housekeeping
 - *Course syllabus and information*
4. Some history
 - *Where did AI come from and where is it going?*
5. Linear regression
 - *What is it and why is it important to understand?*
6. Functions
 - *Some mathematics: what is a function?*
7. Optimization
 - *How do you minimize a function?*
8. Gradients
 - *Some mathematics: what is a gradient?*
9. Gradient descent
 - *Finding the best-fit line*
10. Q&A

About me



Senior Associate Consultant · Radiation Oncology



Senior Researcher · Microsoft Research



Ph.D. in Machine Learning · Cambridge University



Research Scientist Intern · Google DeepMind

More info: andrewfoongyk.github.io

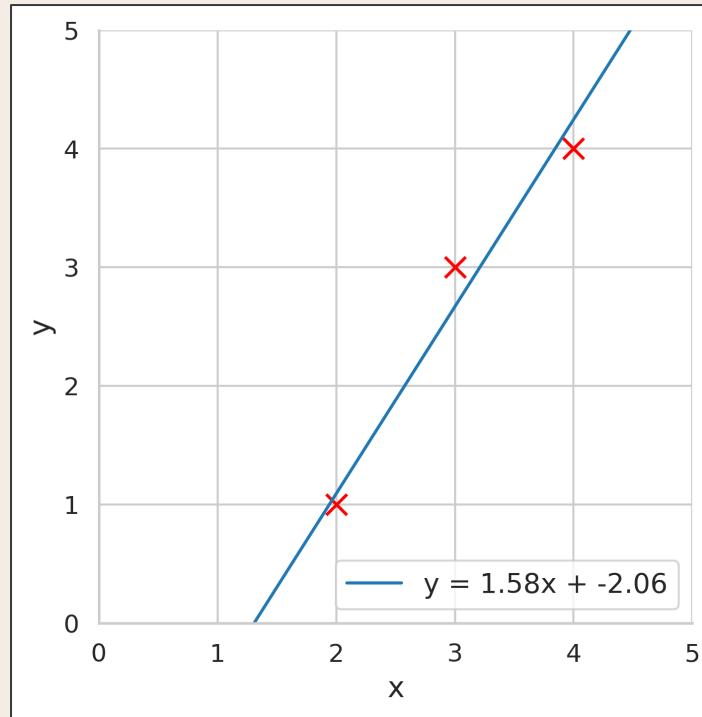
What is AI?

A bird's eye view of the field

What is AI?

Artificial
Intelligence
SLIPPERY
TERM

What is AI?

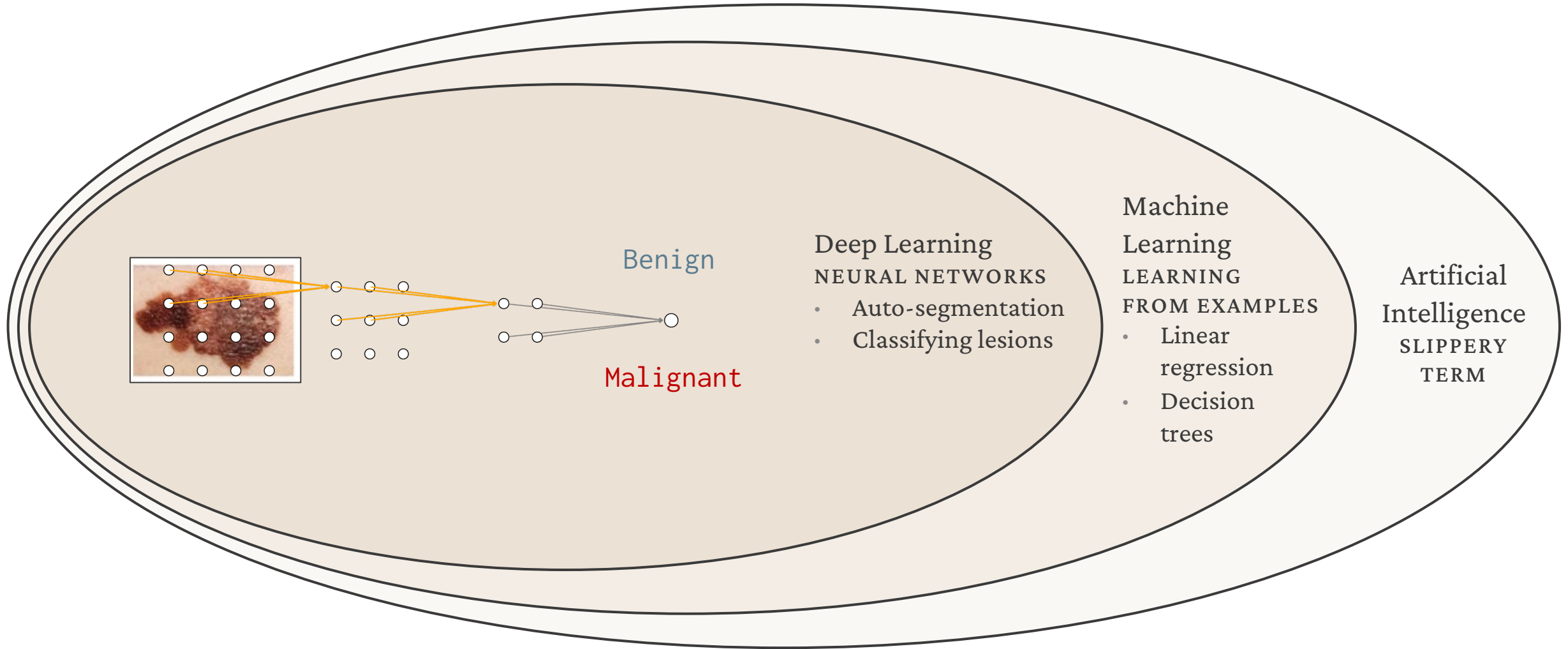


Machine Learning
LEARNING FROM EXAMPLES

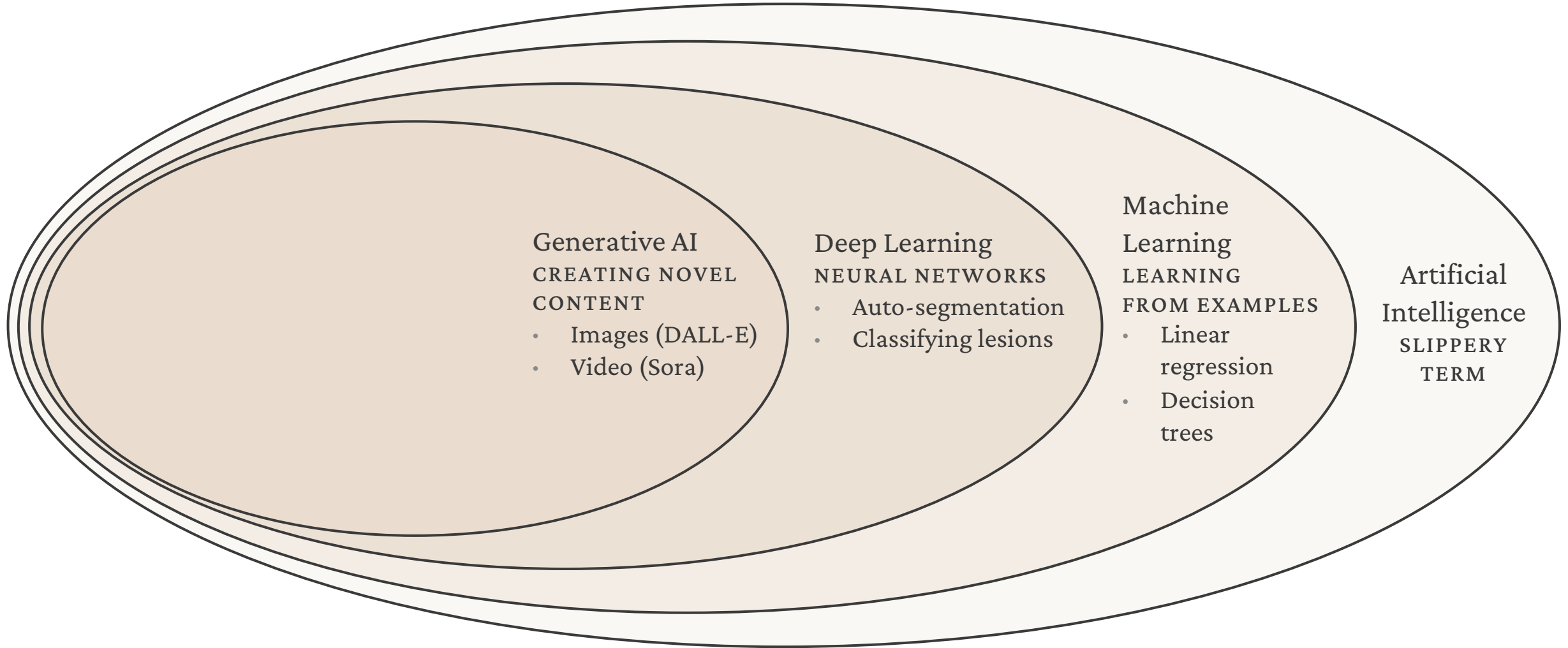
- Linear regression
- Decision trees

Artificial Intelligence
SLIPPERY TERM

What is AI?



What is AI?



What is AI?

Tiny potato kings wearing majestic crowns, sitting on thrones, overseeing their vast potato kingdom filled with potato subjects and potato castles.

Artificial
Intelligence
SLIPPERY
TERM

What is AI?



Generative AI CREATING NOVEL CONTENT

- Images (DALL-E)
- Video (Sora)

Deep Learning NEURAL NETWORKS

- Auto-segmentation
- Classifying lesions

Machine Learning LEARNING FROM EXAMPLES

- Linear regression
- Decision trees

Artificial Intelligence SLIPPERY TERM

What is AI?

A family of grizzly bears sit at a table, dining on salmon sashimi with chopsticks.

Machine

Artificial
Intelligence
SLIPPERY
TERM

What is AI?



Generative AI
CREATING NOVEL
CONTENT
Images (DALL-E)
Video (Sora)

Deep Learning
NEURAL NETWORKS

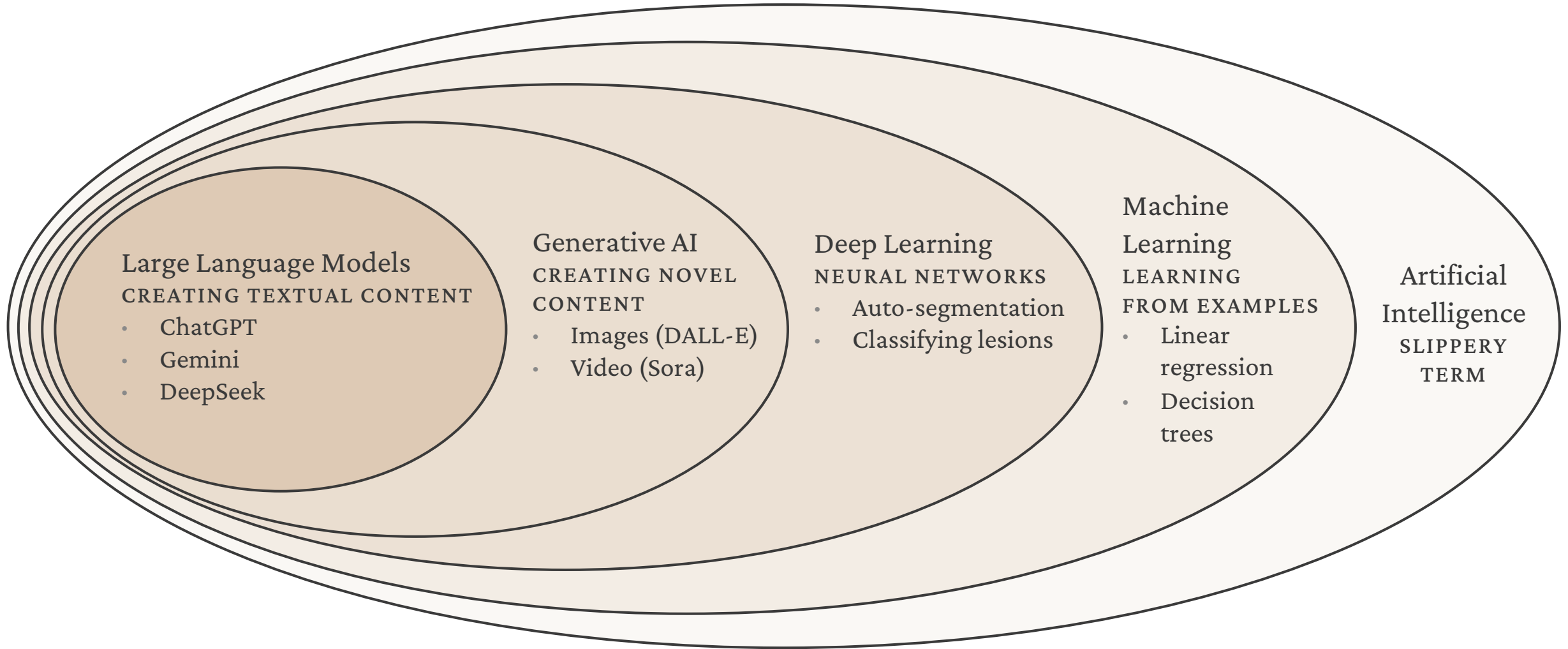
- Auto-segmentation
- Classifying lesions

Machine Learning
LEARNING FROM EXAMPLES

- Linear regression
- Decision trees

Artificial Intelligence
SLIPPERY TERM

What is AI?



What is AI?

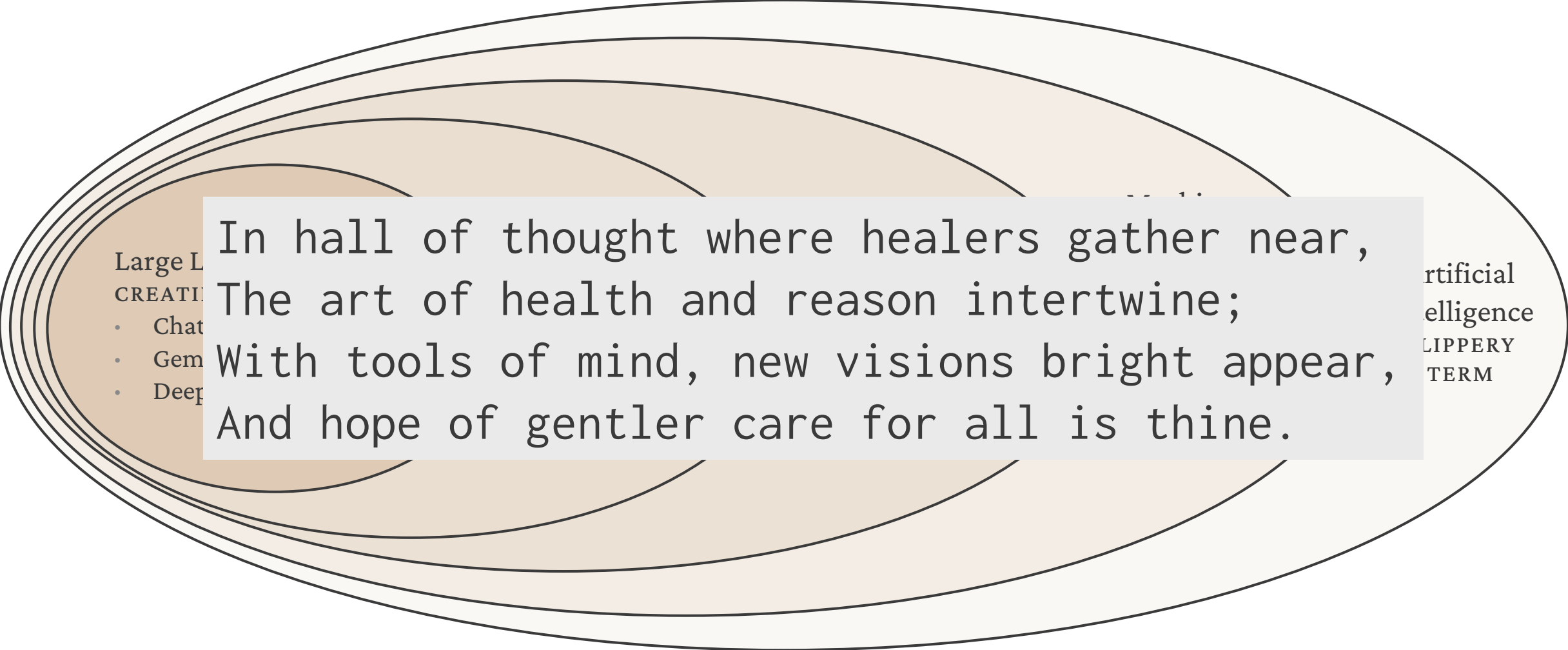
Write a Shakespearean poem
about a lecture course on AI
for healthcare.

Large Language
CREATING

- ChatGPT
- Gemini
- DeepSeek

Artificial
Intelligence
SLIPPERY
TERM

What is AI?



In hall of thought where healers gather near,
The art of health and reason intertwine;
With tools of mind, new visions bright appear,
And hope of gentler care for all is thine.

Large L
CREATI
• Chat
• Gem
• Deep

rtificial
elligence
LIPPERY
TERM

What is AI?

Article

Large language models encode clinical knowledge

<https://doi.org/10.1038/s41586-023-06291-2>

Received: 25 January 2023

Accepted: 5 June 2023

Published online: 12 July 2023

Open access

 Check for updates

Karan Singhal^{1,4,5}, Shekoofeh Azizi^{1,4,5}, Tao Tu^{1,4}, S. Sara Mahdavi¹, Jason Wei¹, Hyung Won Chung¹, Nathan Scales¹, Ajay Tanwani¹, Heather Cole-Lewis¹, Stephen Pfohl¹, Perry Payne¹, Martin Seneviratne¹, Paul Gamble¹, Chris Kelly¹, Abubakr Babiker¹, Nathanael Schärli¹, Aakanksha Chowdhery¹, Philip Mansfield¹, Dina Demner-Fushman², Blaise Agüera y Arcas¹, Dale Webster¹, Greg S. Corrado¹, Yossi Matias¹, Katherine Chou¹, Juraj Gottweis¹, Nenad Tomasev², Yun Liu¹, Alvin Rajkomar¹, Joelle Barral¹, Christopher Semturs¹, Alan Karthikesalingam^{1,5,6} & Vivek Natarajan^{1,5,6}

Large language models (LLMs) have demonstrated impressive capabilities, but the bar for clinical applications is high. Attempts to assess the clinical knowledge of models typically rely on automated evaluations based on limited benchmarks. Here, to address these limitations, we present MultiMedQA, a benchmark combining six existing medical question answering datasets spanning professional medicine, research and consumer queries and a new dataset of medical questions searched online, HealthSearchQA. We propose a human evaluation framework for model answers along multiple axes including factuality, comprehension, reasoning, possible harm and bias. In addition, we evaluate Pathways Language Model¹ (PaLM, a 540-billion parameter LLM) and its instruction-tuned variant, Flan-PaLM² on MultiMedQA. Using a combination of prompting strategies, Flan-PaLM achieves state-of-the-art accuracy on every MultiMedQA multiple-choice dataset (MedQA³, MedMCQA⁴, PubMedQA⁵ and Measuring Massive Multitask Language Understanding (MMLU) clinical topics⁶), including 67.6% accuracy on MedQA (US Medical Licensing Exam-style questions), surpassing the prior state of the art by more than 17%. However, human evaluation reveals key gaps. To resolve this, we introduce instruction prompt tuning, a parameter-efficient approach for aligning LLMs to new domains using a few exemplars. The resulting model, Med-PaLM, performs encouragingly, but remains inferior to clinicians. We show that comprehension, knowledge recall and reasoning improve with model scale and instruction prompt tuning, suggesting the potential utility of LLMs in medicine. Our human evaluations reveal limitations of today's models, reinforcing the importance of both evaluation frameworks and method development in creating safe, helpful LLMs for clinical applications.

Deep Learning NEURAL NETWORKS

- Auto-segmentation
- Classifying lesions

- #### Machine Learning FROM EXAMPLES
- Linear regression
 - Decision trees

Artificial
Intelligence
SLIPPERY
TERM

What is AI?

Article

a

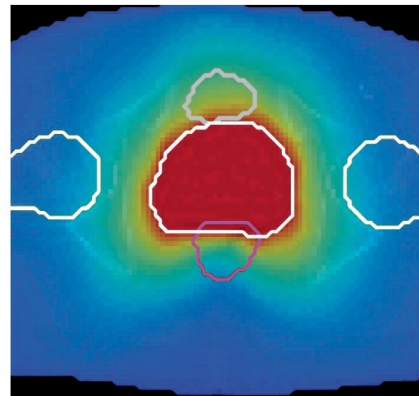
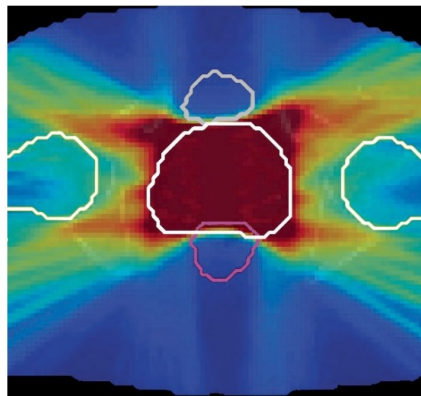
Gradients: The current plan does not meet the criteria for the PTV. The constraints on OARs are currently being met. Recommendation: Increase the importance weight for the PTV.

Gradients: The current weight for the rectum and bladder are relatively low, which is not sufficient to protect the rectum and bladder from receiving higher doses. Recommendation: Slightly increase the importance weight for the bladder and rectum.

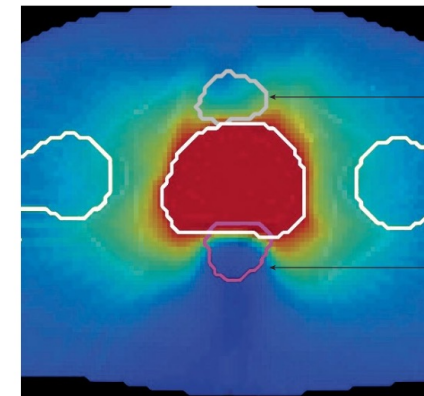
Initialization

Iteration 1

Iteration 5



.....



Bladder

Rectum

Artificial
Intelligence
SLIPPERY
TERM

Yuksekgonul, M., Bianchi, F., Boen, J. et al. Optimizing generative AI by backpropagating language model feedback. Nature 639, 609–616 (2025).
<https://doi.org/10.1038/s41586-025-08661-4>

What is AI?

Called *foundation models*:

- One AI model provides foundation for many tasks.
- Can ask ChatGPT/Gemini/DeepSeek about medicine or English literature or astronomy or anything!
- Contrast *single-task* AI models that are specialized for one purpose.

ial
ence
RY
1

What is AI?

MAGAZINE · A.I.

How DeepSeek erased Silicon Valley's AI lead and wiped \$1 trillion from U.S. markets

BY NICHOLAS GORDON

March 30, 2025 at 8:00 PM EDT



g
ORKS
ntation
esions

Machine Learning
LEARNING FROM EXAMPLES

- Linear regression
- Decision trees

Artificial Intelligence
SLIPPERY TERM

Nicholas Gordon, Fortune, March 30, 2025

What is AI?

The screenshot shows a website header with the title "Generative AI & Automation" and a search bar. Below the header is a navigation menu with links for Home, Generative AI, Automation, Glossary, What's happening, and Gerstner Scholars. The main content area features a large banner with the headline "Transforming healthcare with generative AI and automation". The banner is split into two sections: the left side shows a man working at a computer with a "Learn about Generative AI" button, and the right side shows a woman holding a prosthetic hand with a "Learn about Automation" button. Below the banner, a sub-header reads "THIS IS A DEFINING MOMENT IN MAYO CLINIC'S HISTORY", followed by the text: "Together, we are building the most trusted generative AI and automation solutions to benefit patients worldwide."

Generative AI & Automation

Search this site

Home Generative AI Automation Glossary What's happening Gerstner Scholars

Transforming healthcare with generative AI and automation

Learn about Generative AI

Learn about Automation

THIS IS A DEFINING MOMENT IN MAYO CLINIC'S HISTORY

Together, we are building the most trusted generative AI and automation solutions to benefit patients worldwide.

Mayo Clinic is betting that AI will revolutionize healthcare

What is AI?

Deep learning has already revolutionized the practice of science



JOHN HOPFIELD
Physics
Princeton University

GEOFFREY HINTON
Physics
University of Toronto



DEMIS HASSABIS
Chemistry
Google DeepMind

JOHN JUMPER
Chemistry
Google DeepMind

2024 NOBEL PRIZES

Goals & Philosophy

What is this course about?

Goals & philosophy

- Many of us use/will use AI, but are not AI scientists.
 - ChatGPT, Auto-segmentation...
 - Black-box model:



- Why would a *healthcare professional* need to know inner workings?
 - **CRITICAL THINKING:** When will it work/fail? Don't be passive consumer.
 - **INNOVATION:** Identify promising applications.
 - **CONFIDENCE:** Distinguish hype from reality, call out nonsense.
 - **SCIENTIFIC CURIOSITY:** One of the greatest engineering achievements.

Goals & philosophy

- The world is awash with with talk of AI.
- What used to be a niche technical discipline is now the stuff of LinkedIn influencers.

OVERALL GOAL:

Go beyond the hype and understand what AI actually does on a technical level.

Goals & philosophy

1. Technical understanding of AI.
2. AI intuition
 - What's *easy* and what's *difficult*.
 - *Whether* to apply AI to your problem.
 - What *kind* of AI to apply.
 - How much *data* you'll need.
 - When is AI going to *fail*?
3. AI terminology
 - Deep neural network
 - Training / learning
 - Foundation model
 - GPT
 - Embeddings
 - U-Net
 - Etc.

Goals & philosophy

- Fundamental language of AI is *mathematics*.

“*WHEN YOU CAN MEASURE* what you are speaking about, and express it in numbers, you know something about it, *when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind*; it may be the beginning of knowledge, but you have scarcely, in your thoughts advanced to the stage of science.”

— LORD KELVIN

- Using AI without knowing any underlying math is like practicing medicine without knowing any underlying biology.
- Could probably get by, but won't ever contribute anything *new*.

Goals & philosophy

- Fundamental language of AI is mathematics.
 - All mathematics needed will be explained.
 - Significantly less than a typical biostatistics course.
- We'll use:
 - *Functions* (machine that eats numbers and spits numbers out):

$$f(x) = 5x + 4$$

- *Derivatives* (will recap):

$$\frac{d}{dx} f(x) = 5$$

“Everything should be as simple as it can be, but not simpler.”

— ALBERT EINSTEIN

Housekeeping

Course structure and information

Course syllabus

Fundamentals

1. Background & linear regression
2. From single neurons to neural networks
3. Generalization & learning algorithms

Imaging

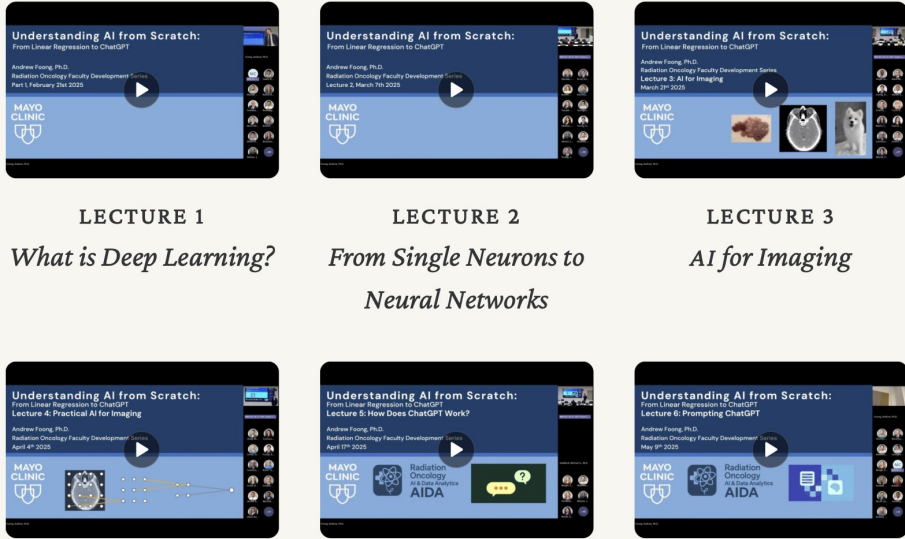
4. Image data and image classification
5. Convolutional neural networks
6. U-nets

Language Models

7. Representing language & generating text
8. LLM conversations and prompting
9. Retrieval-augmented generation and agentic AI
10. The transformer architecture

Additional materials

- *Understanding AI from Scratch: from Linear Regression to ChatGPT*
 - Six-part faculty development series given to Radiation Oncology department.
 - Available at: andrewfoongyk.github.io/ai-from-scratch

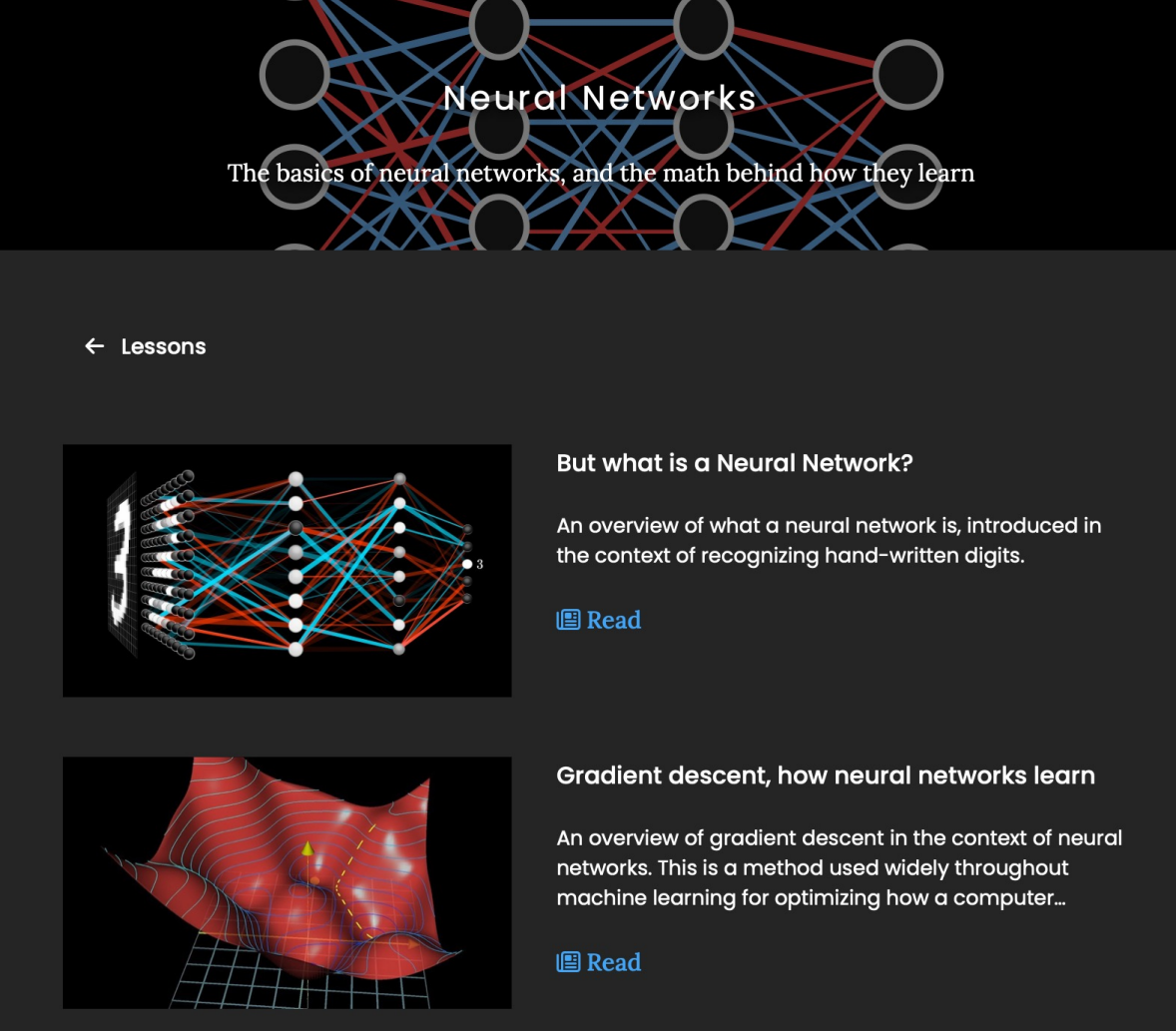


The image displays six video lecture thumbnails arranged in a 2x3 grid. Each thumbnail features a play button in the center and a title below it. The thumbnails are as follows:

- LECTURE 1**
What is Deep Learning?
- LECTURE 2**
From Single Neurons to Neural Networks
- LECTURE 3**
AI for Imaging
- LECTURE 4**
Practical AI for Imaging
- LECTURE 5**
How Does ChatGPT Work?
- LECTURE 6**
Prompting ChatGPT

Additional materials

- *3Blue1Brown*:
 - Mathematical education YouTube channel.
- Ten-part series on AI:
 - 3blue1brown.com/topics/neural-networks



Neural Networks
The basics of neural networks, and the math behind how they learn

← Lessons

But what is a Neural Network?
An overview of what a neural network is, introduced in the context of recognizing hand-written digits.
[Read](#)

Gradient descent, how neural networks learn
An overview of gradient descent in the context of neural networks. This is a method used widely throughout machine learning for optimizing how a computer...
[Read](#)

Additional materials

- *Michael Nielsen's free e-book:*
 - neuralnetworksanddeeplearning.com
 - Concise and well-written.
 - Focuses on mathematical intuition.

Neural Networks and Deep Learning

Neural Networks and Deep Learning is a free online book. The book will teach you about:

- Neural networks, a beautiful biologically-inspired programming paradigm which enables a computer to learn from observational data
- Deep learning, a powerful set of techniques for learning in neural networks

Neural Networks and Deep Learning

What this book is about

On the exercises and problems

▶ Using neural nets to recognize handwritten digits

▶ How the backpropagation algorithm works

▶ Improving the way neural networks learn

▶ A visual proof that neural nets can compute any function

Discussion & questions

- *Please ask questions!*
 - *If you have a question, it's likely other students have the same question too.*
 - I will not present content you're not meant to understand.
- PDFs of slides will be made available after each lecture.

Some History

Where did AI come from and where is it going?

Some history

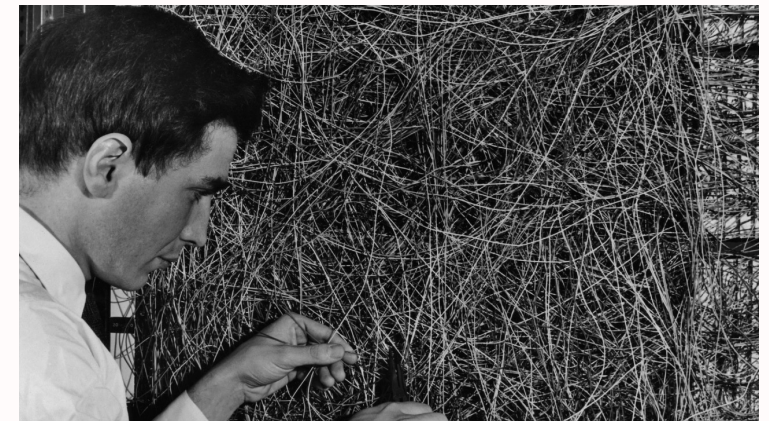
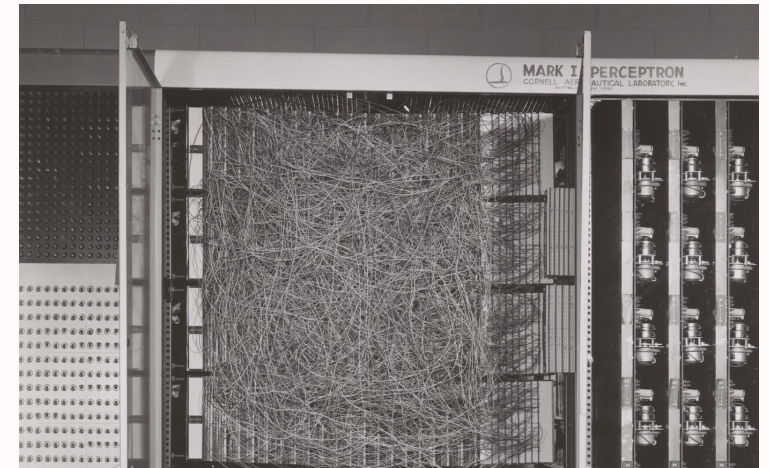
- The goal of creating “thinking machines” is not new.
- **1956** Dartmouth Summer Research Project on Artificial Intelligence:

*“WE PROPOSE THAT A **two-month, ten-man study** of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that **every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.** An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that **a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.**”*

— DARTMOUTH AI PROPOSAL

Some history

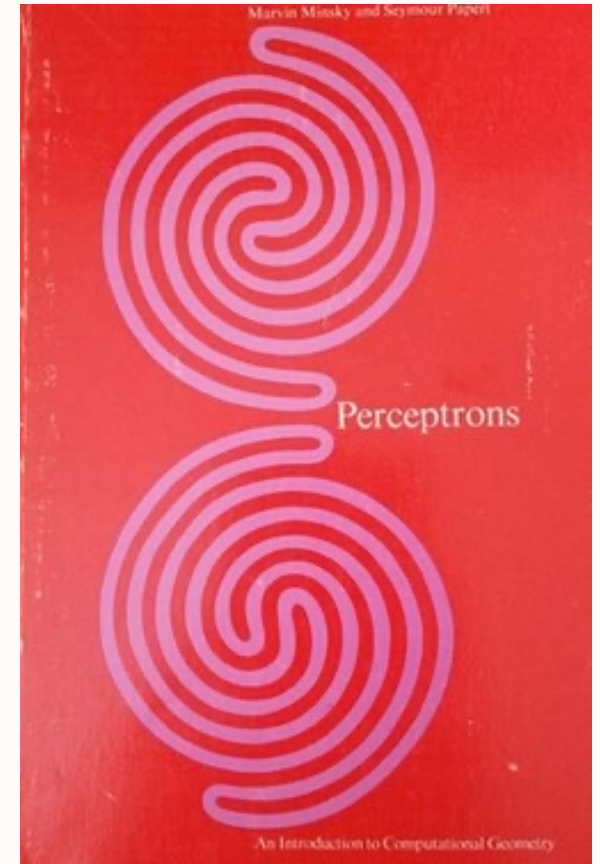
- In 1958, Frank Rosenblatt developed the *Perceptron*.
- Wiring was inspired by the connections of neurons in the brain.
- Used for image recognition.
- Designed to learn directly from experience.
- Many of the basic ideas brought over to today's neural networks.



Frank Rosenblatt and his Perceptron machine,
circa 1960

Some history

- Perceptrons could only recognize certain kinds of patterns.
 - We would call these *linearly separable* patterns.
- A famous book, *Perceptrons* by Marvin Minsky and Seymour Papert proved these limitations.
- Funding for AI and Perceptrons dried up: *The AI Winter*



Minsky, M., Papert, S. (1969).
Perceptrons: An Introduction to
Computational Geometry.
Cambridge, MA, USA: MIT Press.

Some history

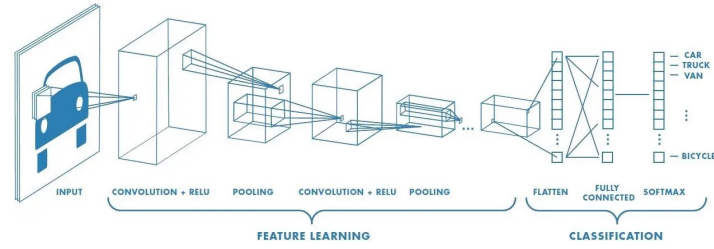
- AI systems were still developed, but very differently from the Perceptron.
- Instead of connections resembling neurons, *strict logical rules* were created.
- Instead of learning directly from experience, *human experts programmed the rules* manually.
- This approach had its successes:
 - In 1996, IBM's Deep Blue defeated Garry Kasparov in chess.
- But there were serious drawbacks:
 - The rules were inevitably brittle.
 - Human experts programming every system was not scalable.



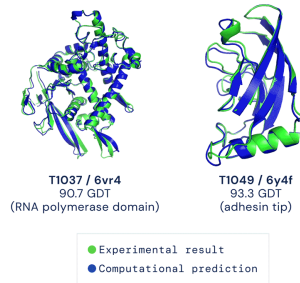
Garry Kasparov vs Deep Blue, 1996

Some history

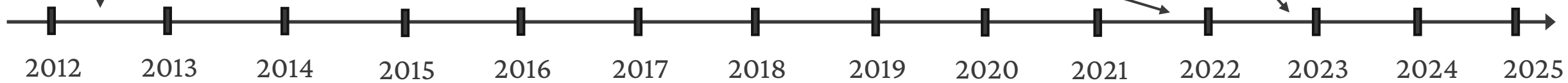
AlexNet:
Image classification



AlphaFold:
Protein structure prediction



ChatGPT:
AI chatbot



AlphaGo:
Playing board games



DALL-E:
Text-to-image generation



Your idea here

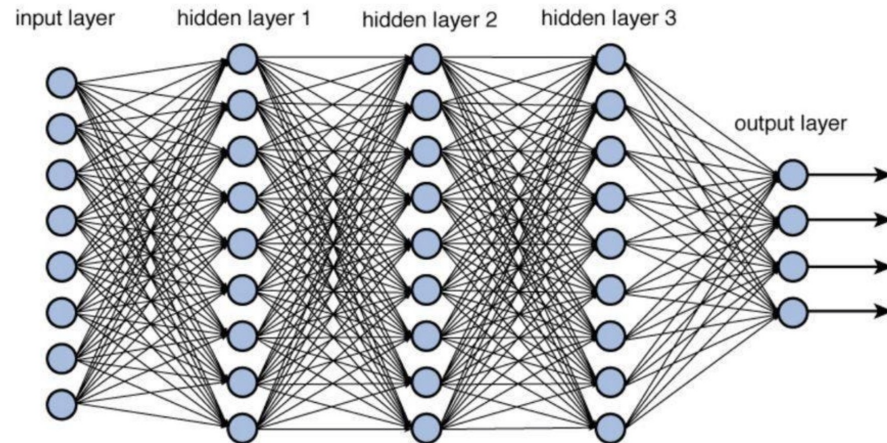


Linear regression

What is it and why is it important to understand?

Deep learning from 40,000 feet

- Deep learning = use of **neural networks**.
- Math functions with millions of numbers: “**parameters/weights**”
- The numbers determine how the neural network behaves.
 1. Start by choosing parameters randomly (garbage predictions).
 2. **Optimizer** automatically adjusts parameters to fit example data.
 3. Apply the function to new data (great predictions, hopefully).

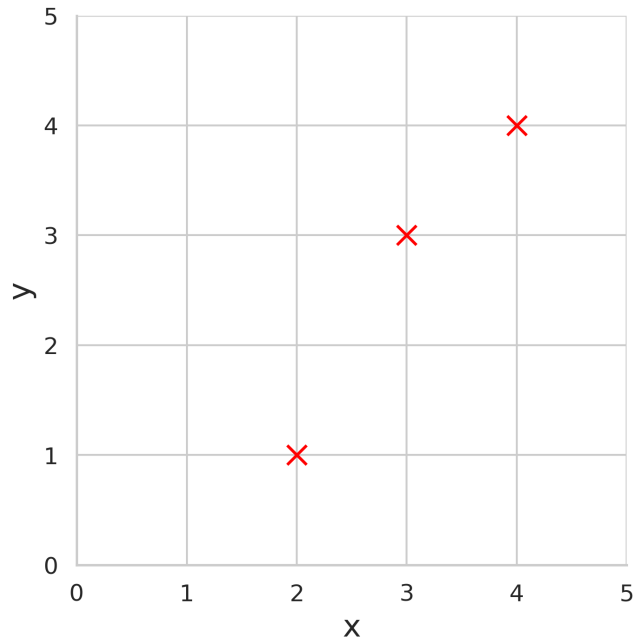


Linear regression

- All these principles can be illustrated with *linear regression*.
 - Linear = “with a straight line”.
 - Regression = “Predict a number from another number”.
- Linear regression can be thought of as a very simple neural network.
- Solving it uses many of the same ideas behind ChatGPT!

Linear regression

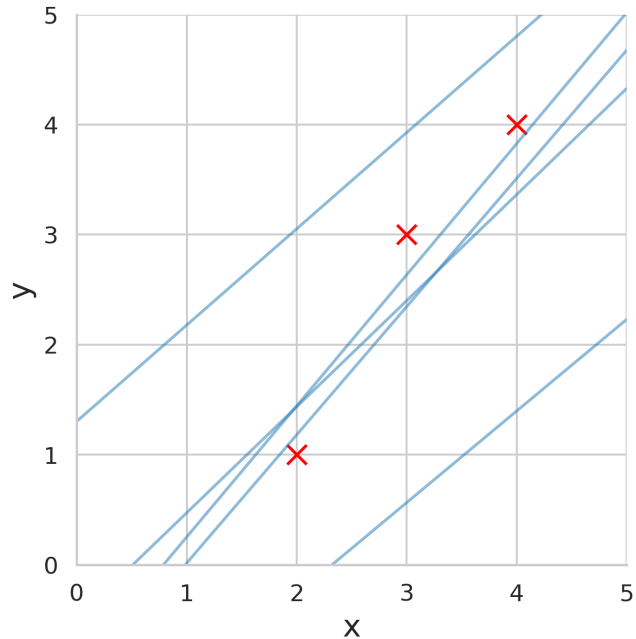
EXAMPLE:



- Given dataset of 3 datapoints:
 - Given any x , want to predict y .
 - How to do this?
- Simplest idea: fit a straight line.
 - Linear regression
 - But points not in straight line!
 - Any straight line will miss *some* points.

Linear regression

EXAMPLE:



- Given dataset of 3 datapoints:
 - Given any x , want to predict y .
 - How to do this?
- Simplest idea: fit a straight line.
 - Linear regression
 - But points not in straight line!
 - Any straight line will miss *some* points.
 - *Which line to choose?*

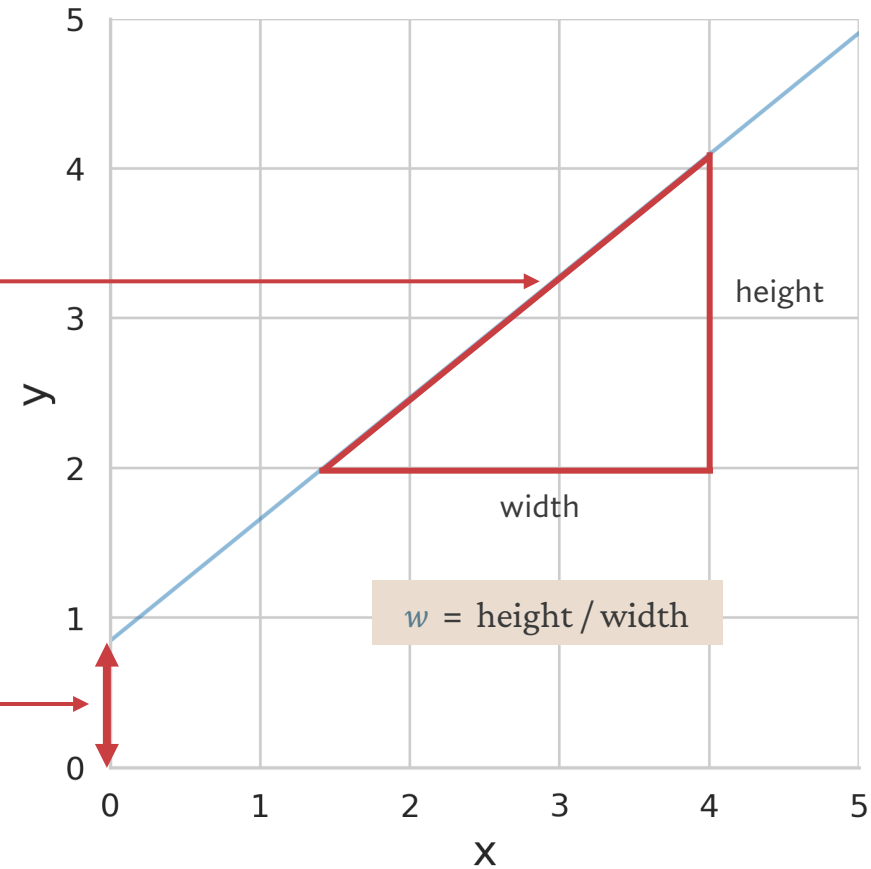
Linear regression parameters

- Need two numbers to define a straight line:

- Slope/gradient: w
- Intercept/bias: b

$$f(x) = wx + b$$

- Once w, b specified, can easily compute $f(x)$ at any x .
- w, b are the parameters/weights.
- Neural networks have them too, but *many* more.
- Reframe problem:
 - Choose best line \rightarrow choose best parameters/weights



Choosing the best-fit line

- Need a rule to choose “best” line.
- No one “correct” answer: we must *invent* a rule.
 - This rule should encapsulate what we care about—it’s *our choice*.
- Define a **LOSS FUNCTION** to measure how far points are from line.
 - **High loss** → line fits data *poorly*.
 - **Low loss** → line fits data *well*.
- Crucial step in all machine learning applications.



Functions

Some mathematics: what is a function?

Recap: functions

- A function is a *mathematical rule* for:
 - Taking a (or multiple) numbers as input.
 - Outputting a (or multiple) numbers.
- Key properties:
 - Behavior is totally specified.
 - Same input \rightarrow same output.

Recap: functions

EXAMPLE:

$$f(x) = x + 1$$

This implies:

$$f(1) = 2$$

$$f(2) = 3$$

$$f(3) = 4$$

...

Recap: functions

EXAMPLE:

$$f(x) = x^2$$

This implies:

$$f(1) = 1$$

$$f(2) = 4$$

$$f(3) = 9$$

...

Recap: functions

EXAMPLE:

$$f(x_1, x_2) = x_1^2 + x_2$$

This implies:

$$f(1, 2) = 1 + 2 = 3$$

$$f(2, 3) = 4 + 3 = 7$$

$$f(3, 1) = 9 + 1 = 10$$

...

Recap: functions

- *Notation:* it can be convenient to arrange multiple numbers into a *vector*.
- Can think of a vector as just a list of numbers.
- Usually written in **boldface**.

EXAMPLE:

$$\mathbf{x} = [x_1, x_2]$$
$$f(\mathbf{x}) = f(x_1, x_2) = x_1^2 + x_2$$

This implies:

$$f(1, 2) = 1 + 2 = 3$$

$$f(2, 3) = 4 + 3 = 7$$

$$f(3, 1) = 9 + 1 = 10$$

...

Choosing a loss function

- Choosing a **LOSS FUNCTION** is an art.
 - *Defines* what a good prediction is.

- Simple and convenient choice: **SQUARED ERROR.**

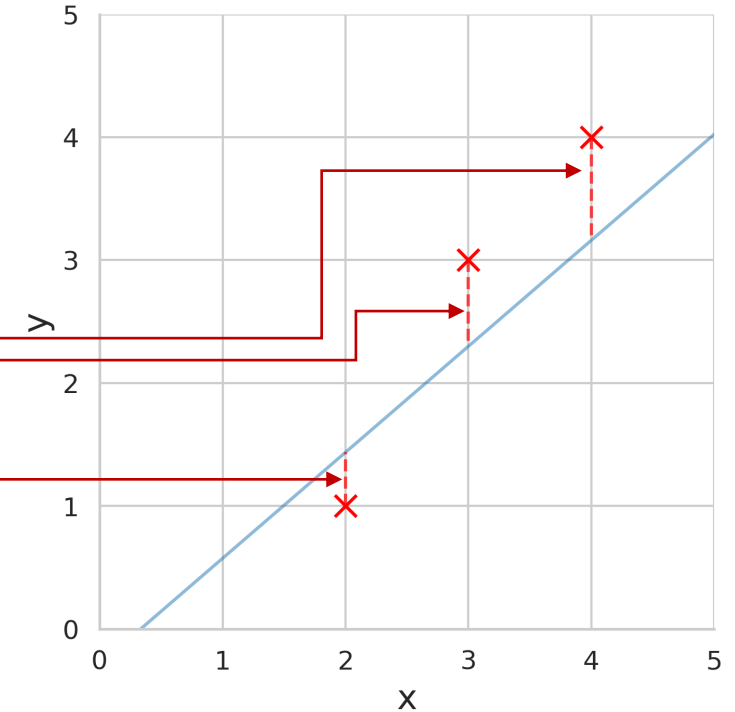
$$L(w, b) = \underbrace{(f(x_1) - y_1)^2}_{\text{error}_1} + \underbrace{(f(x_2) - y_2)^2}_{\text{error}_2} + \underbrace{(f(x_3) - y_3)^2}_{\text{error}_3}$$

- Shorthand notation:

$$L(w, b) = \sum_{n=1}^3 (f(x_n) - y_n)^2$$

- *Idea:* take all differences, square them, add them up.
- Remember $f(x) = wx + b$:

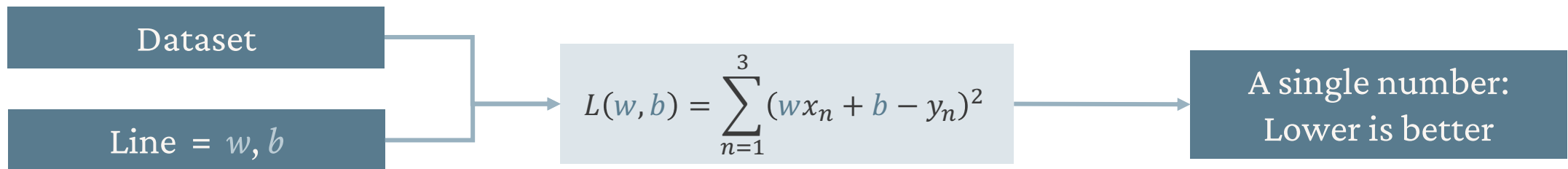
$$L(w, b) = \sum_{n=1}^3 (wx_n + b - y_n)^2$$



Choosing the best-fit line

- Now that we've chosen our loss function, we need to choose the best line.
- The best line: the one that *minimizes* the value of the loss function for our dataset.
- The dataset is fixed, but the line is our choice.

EXAMPLE:



Optimization

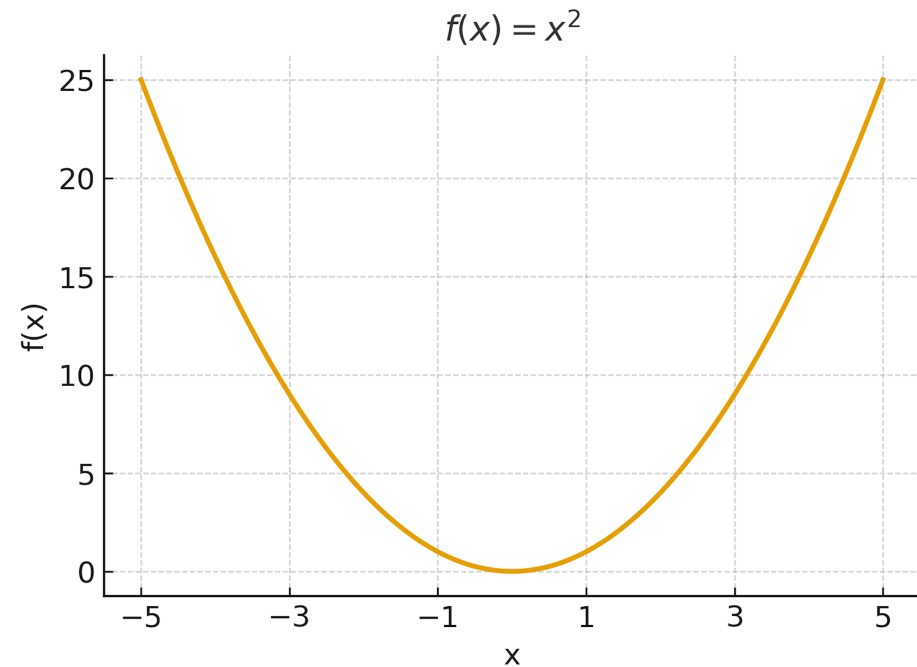
How do you minimize a function?

Optimization

- *Optimization* is the process of finding the input to a function that makes the output as big—or as small—as possible.

EXAMPLE:

- *Find the input x that makes $f(x)$ as small as possible, where $f(x) = x^2$.*
- **ANSWER:** $f(x)$ is minimized when $x = 0$.

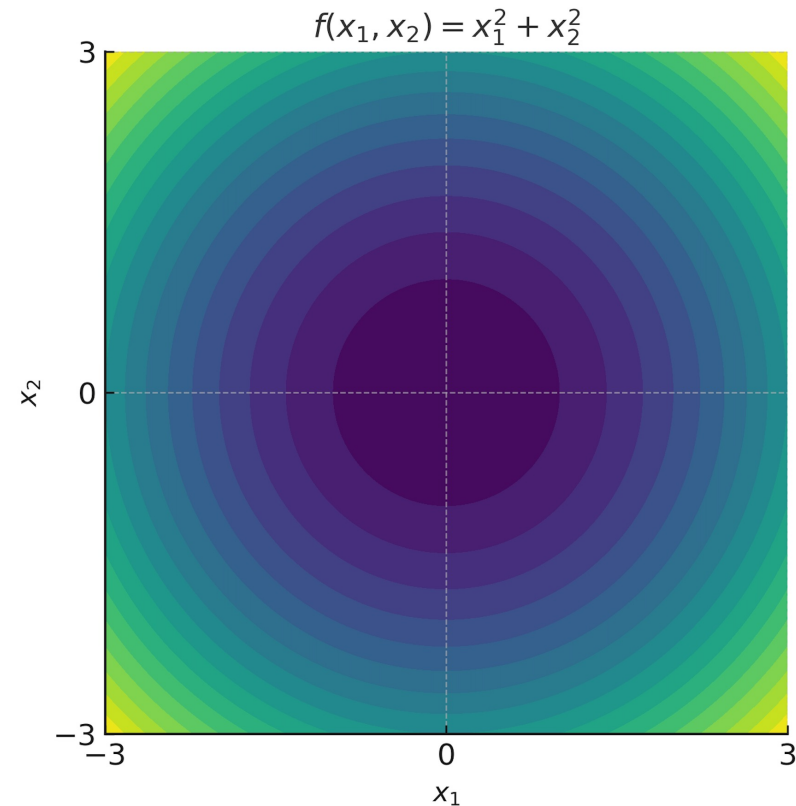


Optimization

- *Optimization* is the process of finding the input to a function that makes the output as big—or as small—as possible.

EXAMPLE:

- Find the input \mathbf{x} that makes $f(\mathbf{x})$ as small as possible, where $f(\mathbf{x}) = x_1^2 + x_2^2$
- ANSWER: $f(\mathbf{x})$ is minimized when $\mathbf{x} = [0, 0]$.

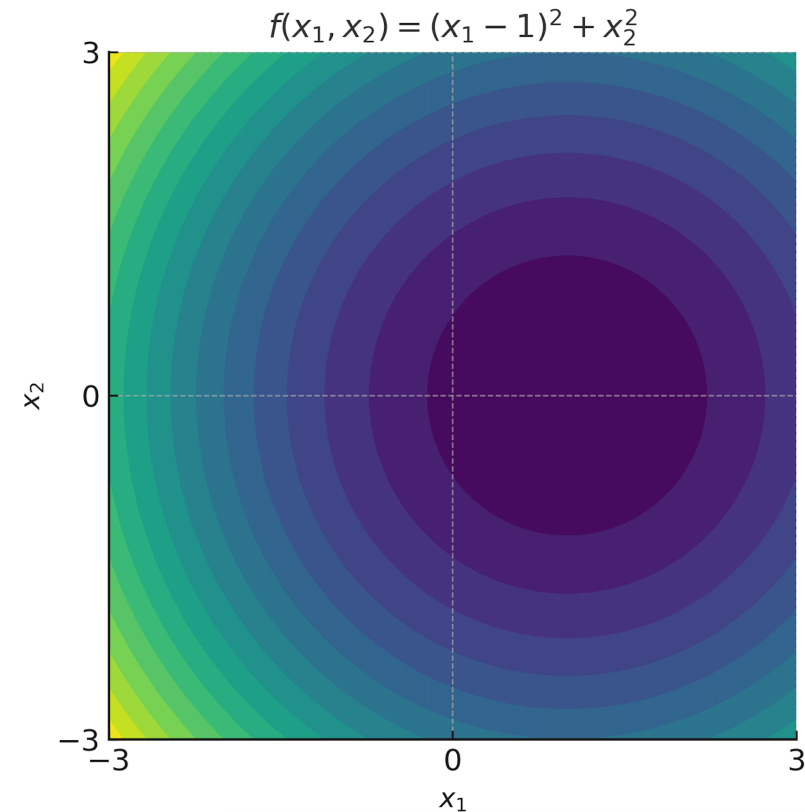


Optimization

- *Optimization* is the process of finding the input to a function that makes the output as big—or as small—as possible.

EXAMPLE:

- Find the input \mathbf{x} that makes $f(\mathbf{x})$ as small as possible, where
$$f(\mathbf{x}) = (x_1 - 1)^2 + x_2^2$$
- ANSWER: $f(\mathbf{x})$ is minimized when $\mathbf{x} = [1, 0]$.

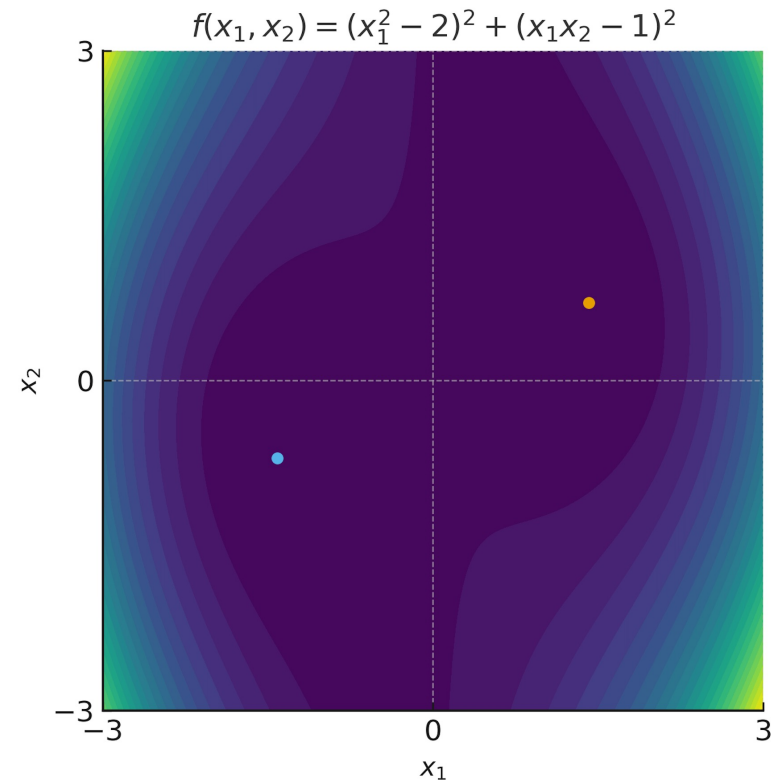


Optimization

- *Optimization* is the process of finding the input to a function that makes the output as big—or as small—as possible.

EXAMPLE:

- Find the input x that makes $f(\mathbf{x})$ as small as possible, where
$$f(\mathbf{x}) = (x_1^2 - 2)^2 + (x_1 x_2 - 1)^2$$
- ANSWER: ???



Optimization

- Not all loss functions can be optimized by looking at the equation.
- Some are too complicated to write down, let alone optimize by hand!
- And if there are too many inputs, e.g.,
 - $f(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$
 - Then *plotting* the function is impossible too.
 - Can't visualize a 7-dimensional space!
- How do we optimize then?
- ANSWER: use **gradients**.

Gradients

Some mathematics: what is a gradient?

Gradients

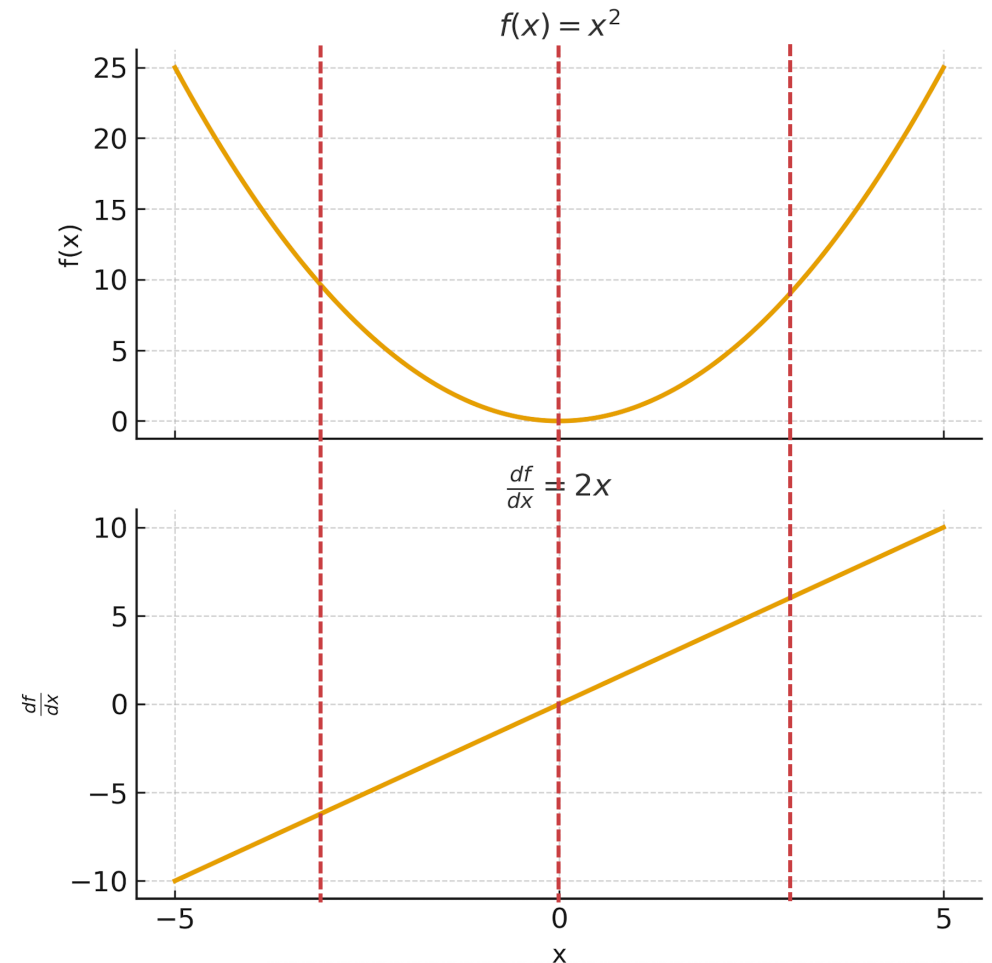
- *Gradient* is another word for *derivative*.
- A function $f(x)$ has a gradient-function called $\frac{df}{dx}(x)$.
- The gradient $\frac{df}{dx}$ is a *function*, just like f .
- That means $\frac{df}{dx}$ takes an input x and transforms it into an output $\frac{df}{dx}(x)$.
- This function has a special interpretation:

When given input x , the function $\frac{df}{dx}$ outputs the value of the *slope* of the function f at the location x .

Gradients

EXAMPLE:

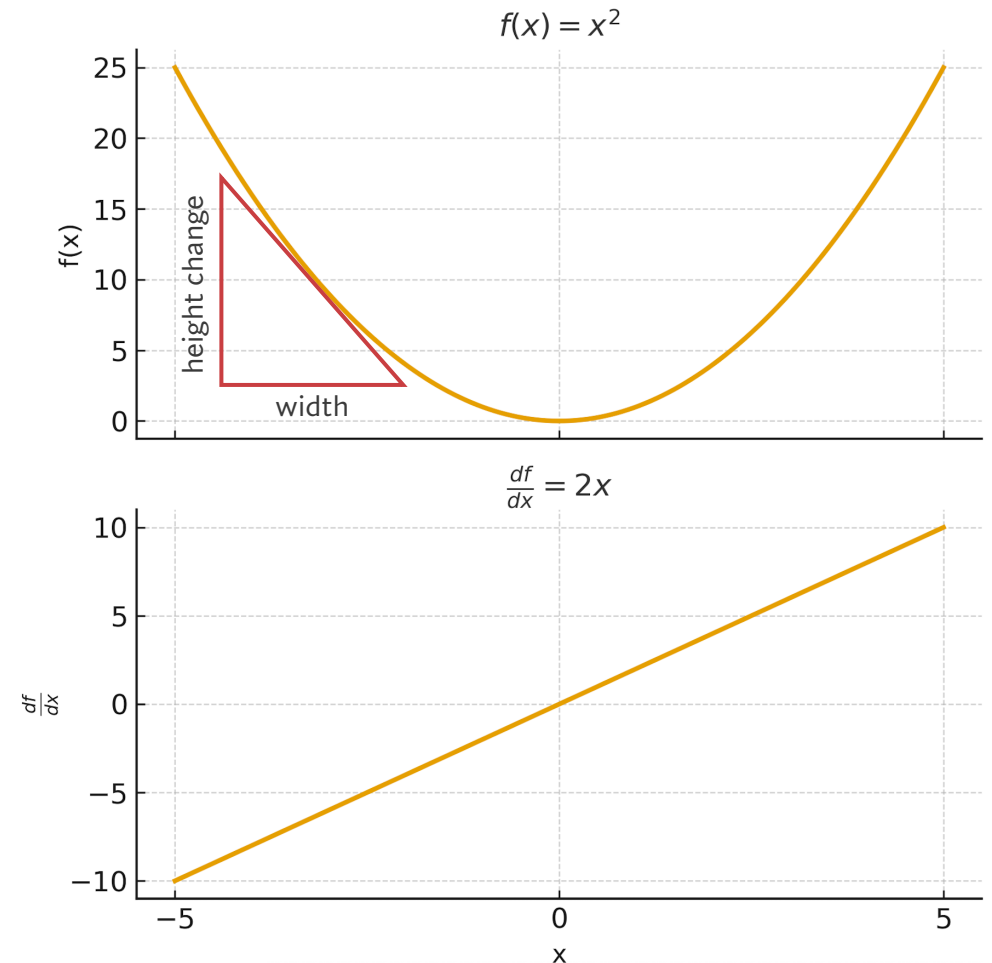
- If $f(x) = x^2$, then $\frac{df}{dx}(x) = 2x$.
- NOTE: $\frac{df}{dx}(x)$ can be easily calculated on a computer using the *backpropagation* algorithm.
- NOTICE:
 - When the slope is downwards, the gradient is *negative*.
 - When the slope is flat, the gradient is *zero*.
 - When the slope is upwards, the gradient is *positive*.



Gradients

EXAMPLE:

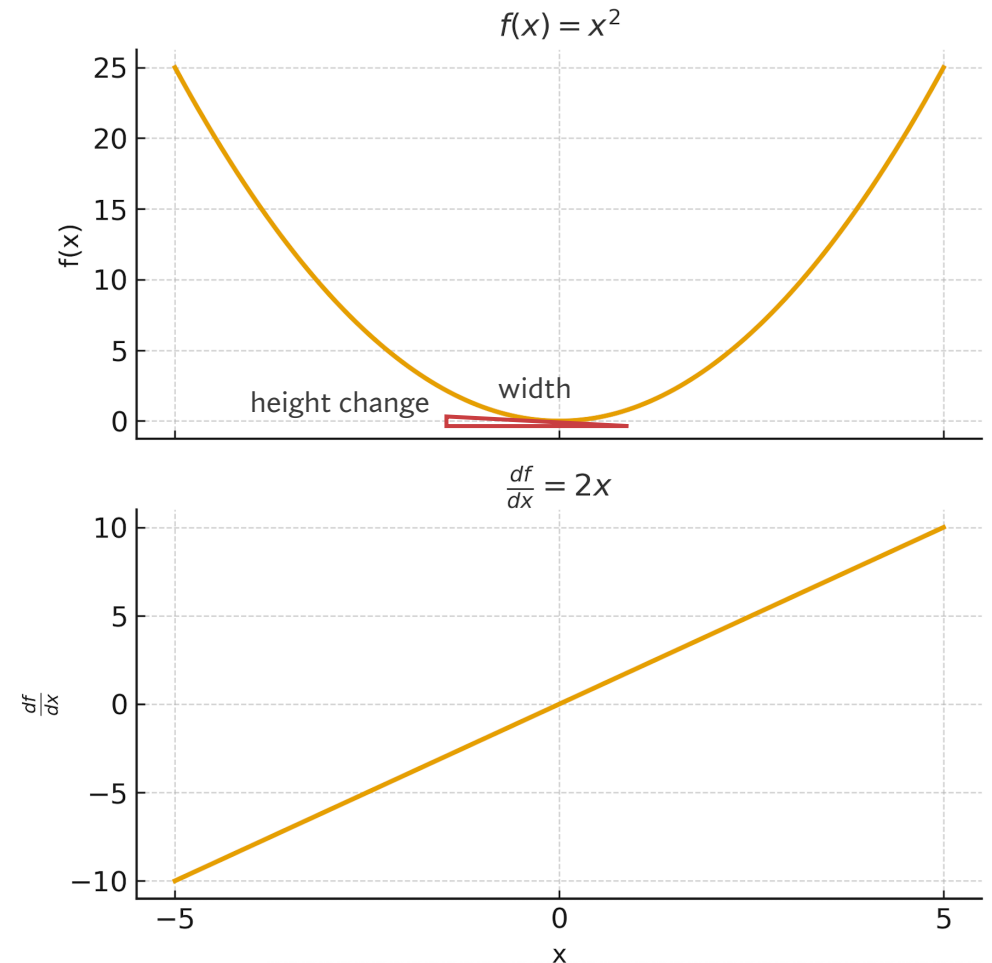
- If $f(x) = x^2$, then $\frac{df}{dx}(x) = 2x$.
- NOTE: $\frac{df}{dx}(x)$ can be easily calculated on a computer using the *backpropagation* algorithm.
- NOTICE:
 - When the slope is downwards, the gradient is *negative*.
 - When the slope is flat, the gradient is *zero*.
 - When the slope is upwards, the gradient is *positive*.



Gradients

EXAMPLE:

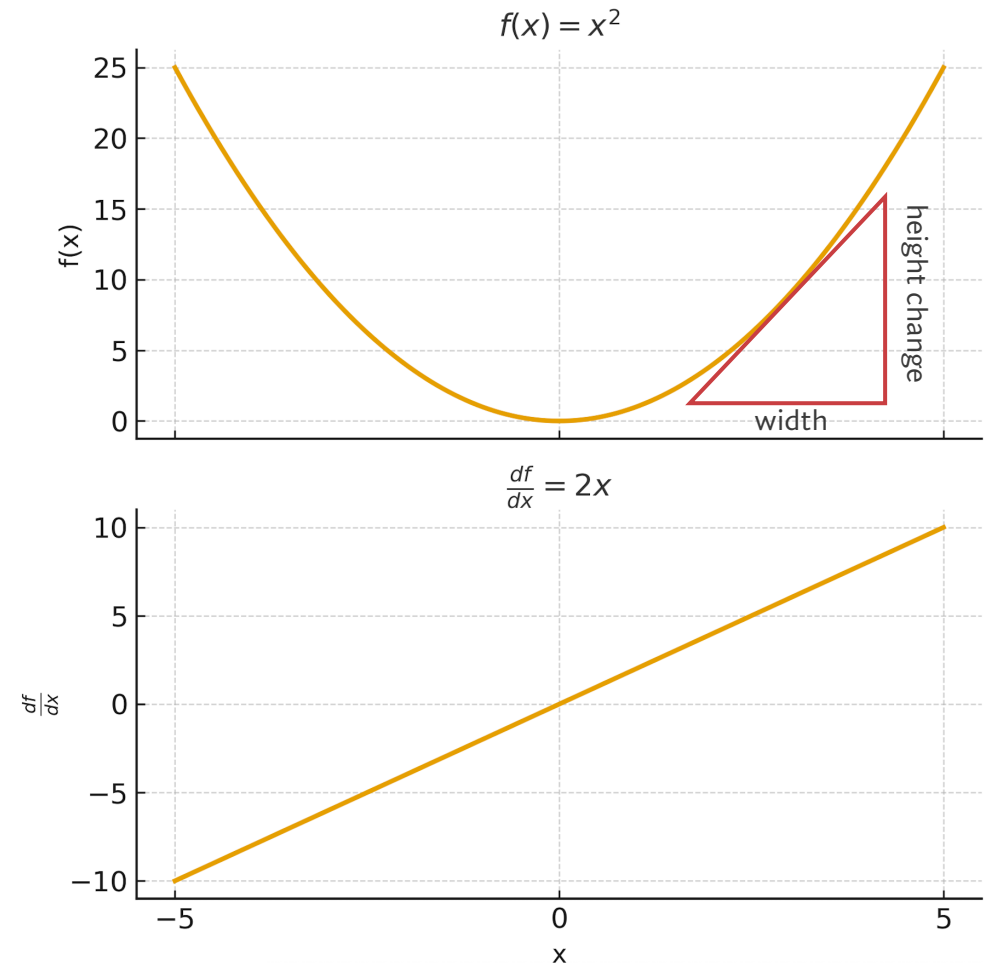
- If $f(x) = x^2$, then $\frac{df}{dx}(x) = 2x$.
- NOTE: $\frac{df}{dx}(x)$ can be easily calculated on a computer using the *backpropagation* algorithm.
- NOTICE:
 - When the slope is downwards, the gradient is *negative*.
 - When the slope is flat, the gradient is *zero*.
 - When the slope is upwards, the gradient is *positive*.



Gradients

EXAMPLE:

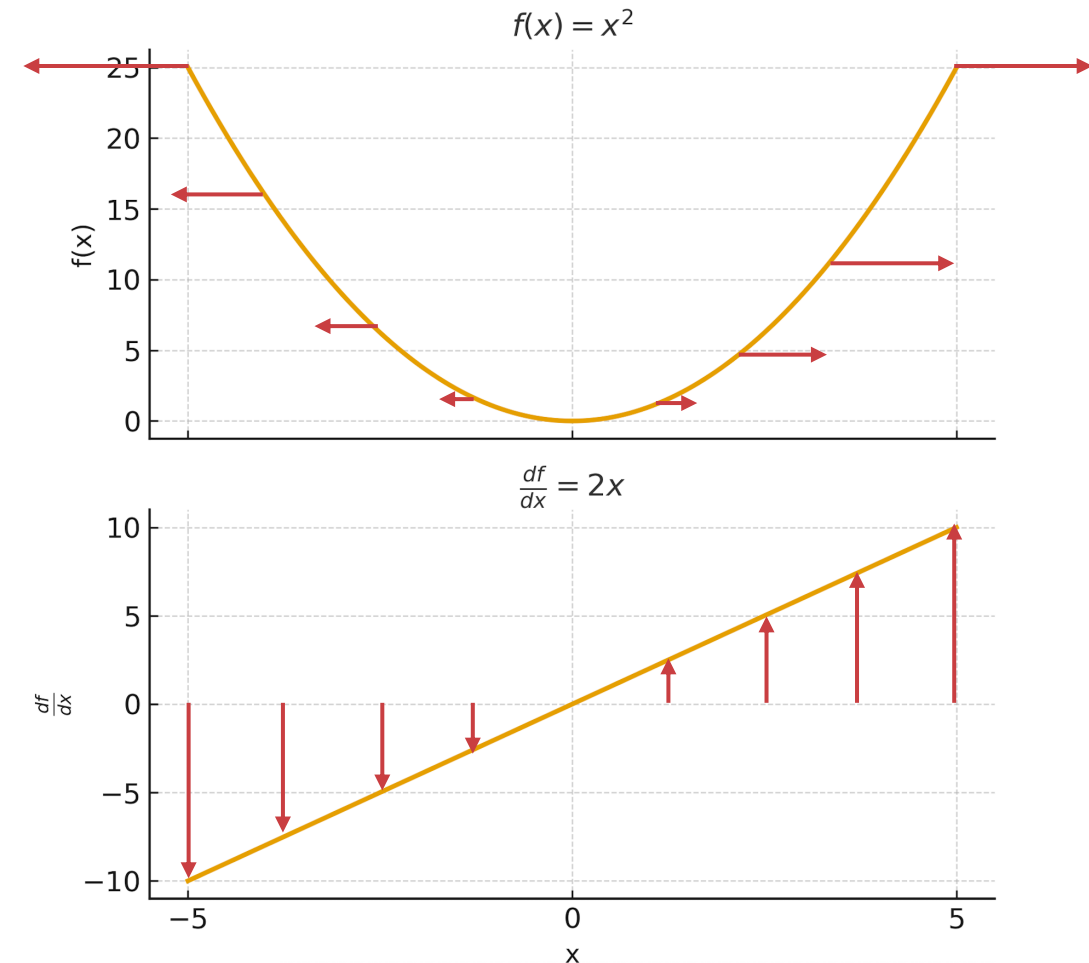
- If $f(x) = x^2$, then $\frac{df}{dx}(x) = 2x$.
- NOTE: $\frac{df}{dx}(x)$ can be easily calculated on a computer using the *backpropagation* algorithm.
- NOTICE:
 - When the slope is downwards, the gradient is *negative*.
 - When the slope is flat, the gradient is *zero*.
 - When the slope is upwards, the gradient is *positive*.



Gradients

EXAMPLE:

- If $f(x) = x^2$, then $\frac{df}{dx}(x) = 2x$.
- NOTE: $\frac{df}{dx}(x)$ can be easily calculated on a computer using the *backpropagation* algorithm.
- NOTICE:
 - When the slope is downwards, the gradient is *negative*.
 - When the slope is flat, the gradient is *zero*.
 - When the slope is upwards, the gradient is *positive*.
 - The gradient DIRECTION tells you *which way to go to increase the value of f*.
 - The gradient MAGNITUDE tells you *how quickly f is increasing in that direction*.



Gradient descent

Finding the best-fit line

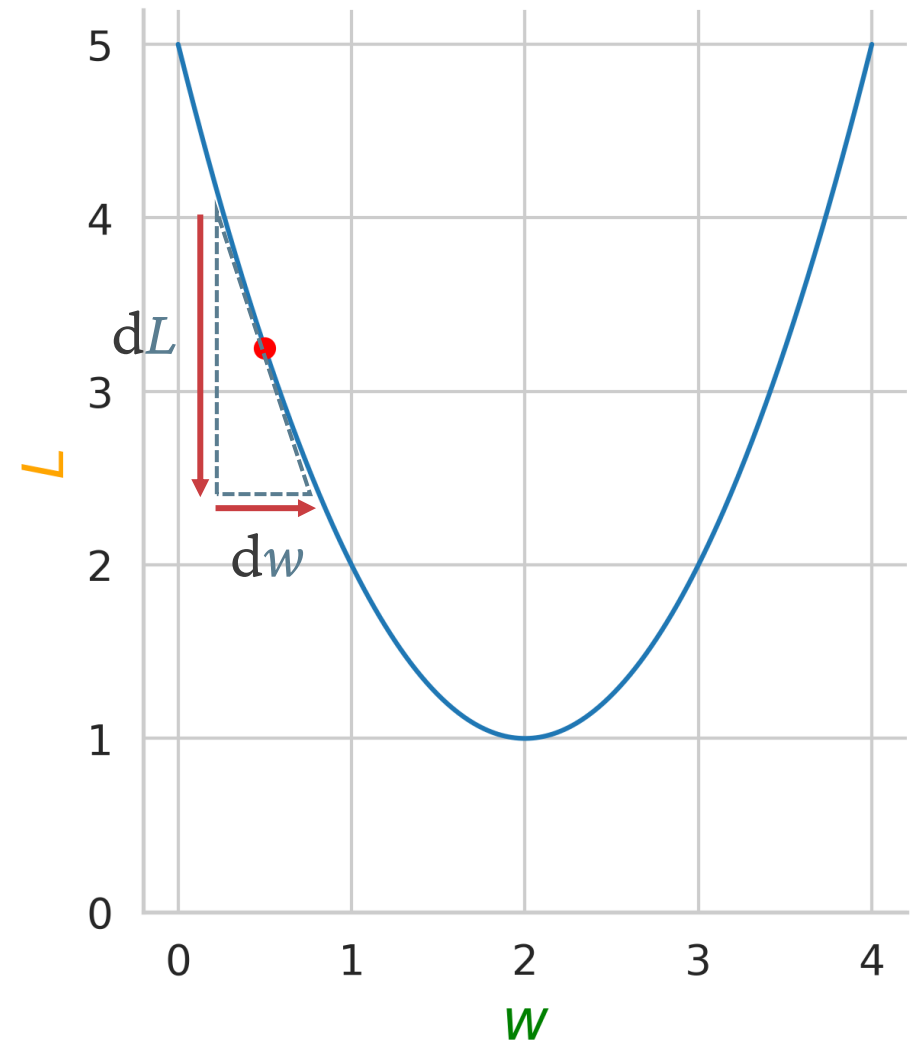
Gradient descent

- *Gradient descent* is an optimization algorithm that uses this information from the gradient function to automatically optimize a loss function.
- **PROCEDURE:**
 1. Choose w, b *randomly* (terrible loss!).
 2. Calculate **derivative/gradient** of $L(w, b)$ —easy on a computer.
 3. Adjust w, b by small amount in (opposite) direction of gradient.
 4. Repeat 2–3 until $L(w, b)$ is low (good fit!).
- Gradient descent is how *all* neural networks learn!

Gradient descent: toy example

EXAMPLE:

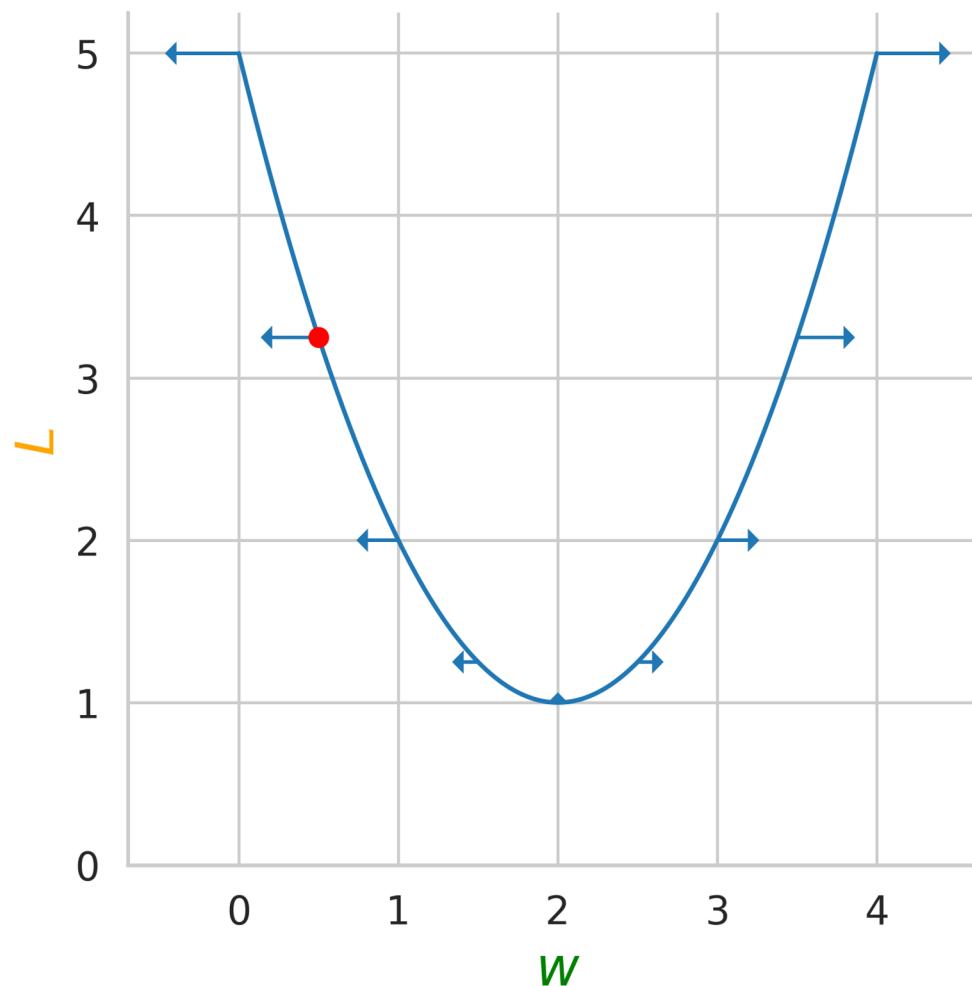
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- **GRADIENT DESCENT:**
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$



Gradient descent: toy example

EXAMPLE:

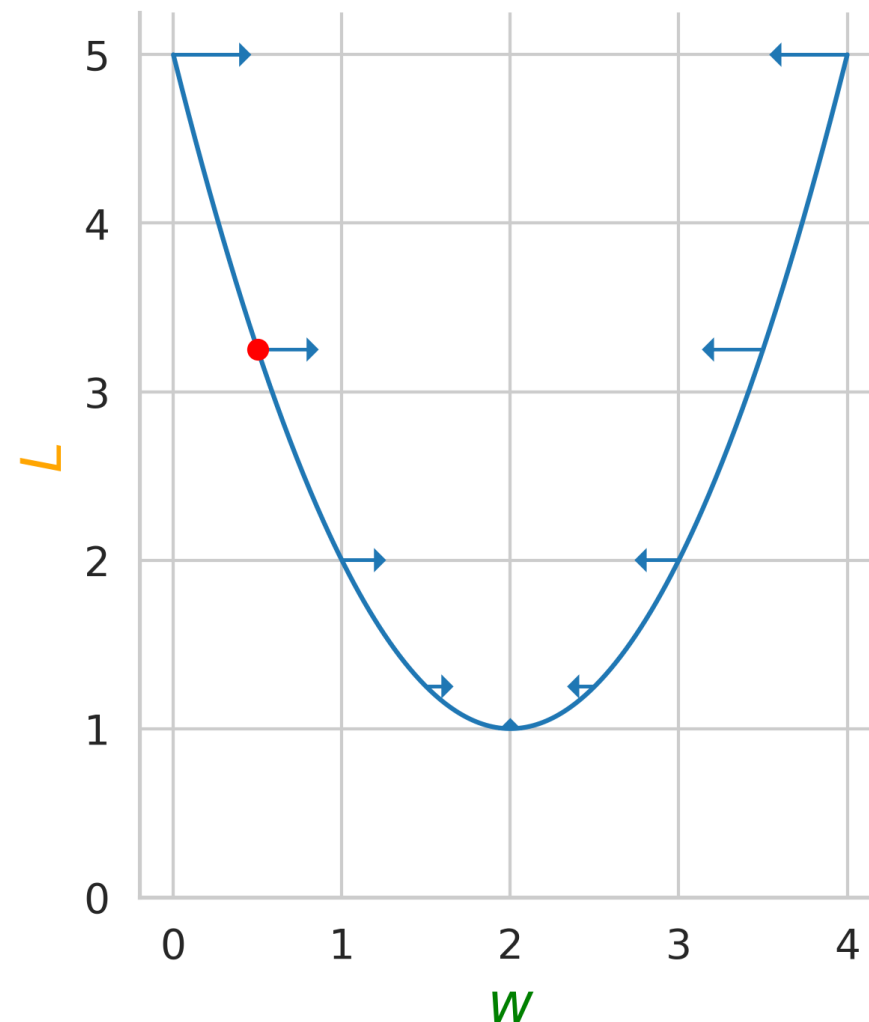
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- **GRADIENT DESCENT:**
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 - Positive if slope is up.
 - Negative if slope is down.
 - Large value if slope is steep.
 - Small value if slope is gentle.
 - Tells you direction and how fast to go to increase L .



Gradient descent: toy example

EXAMPLE:

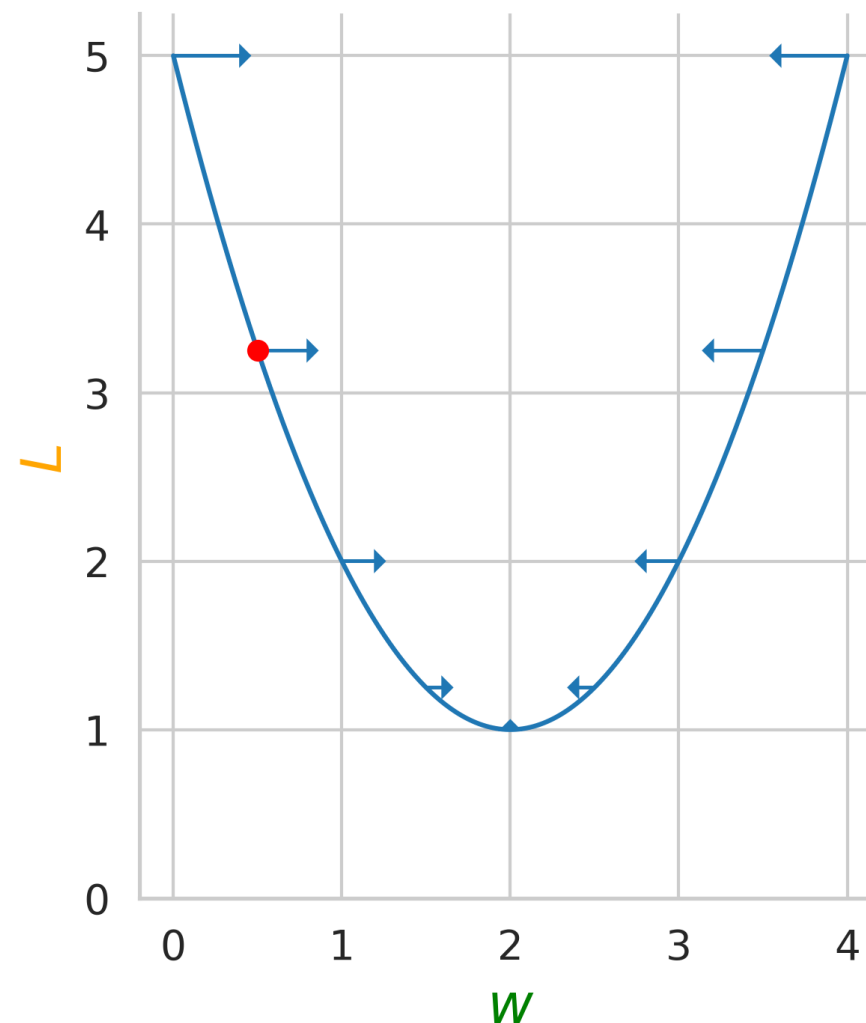
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- **GRADIENT DESCENT:**
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 - Positive if slope is up.
 - Negative if slope is down.
 - Large value if slope is steep.
 - Small value if slope is gentle.
 - Tells you direction and how fast to go to increase L .
 3. Move w a little in *opposite* direction of gradient.



Gradient descent: toy example

EXAMPLE:

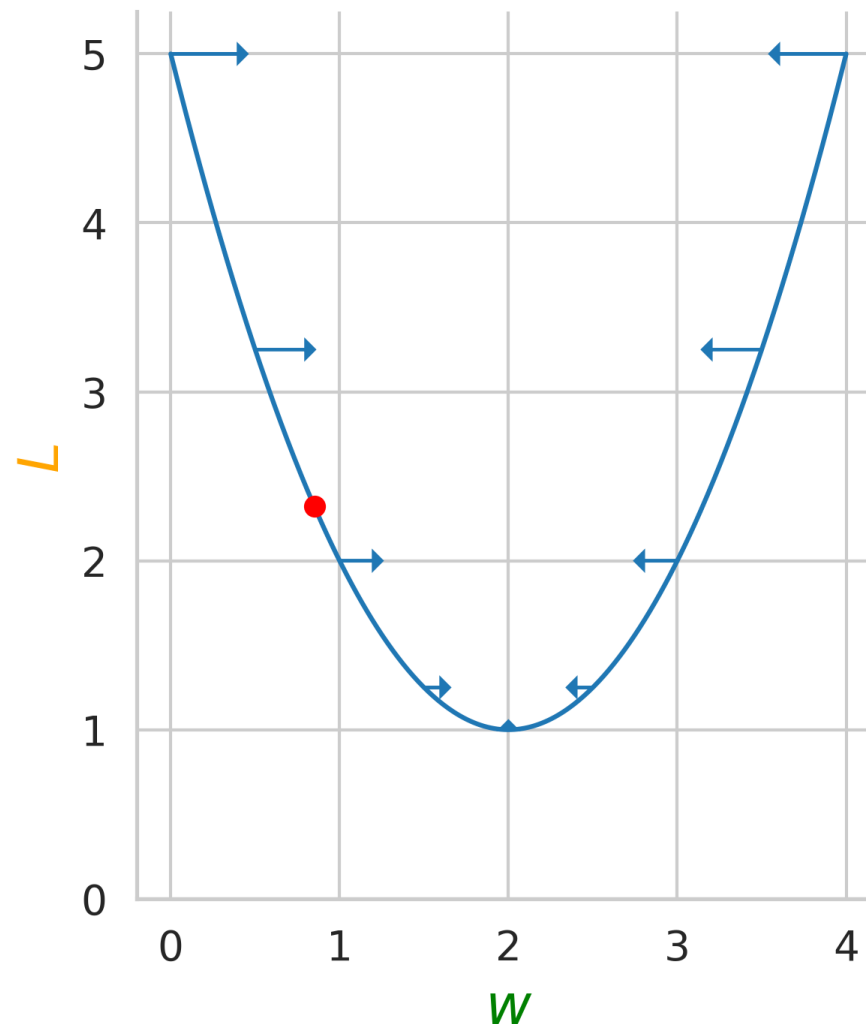
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- **GRADIENT DESCENT:**
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.



Gradient descent: toy example

EXAMPLE:

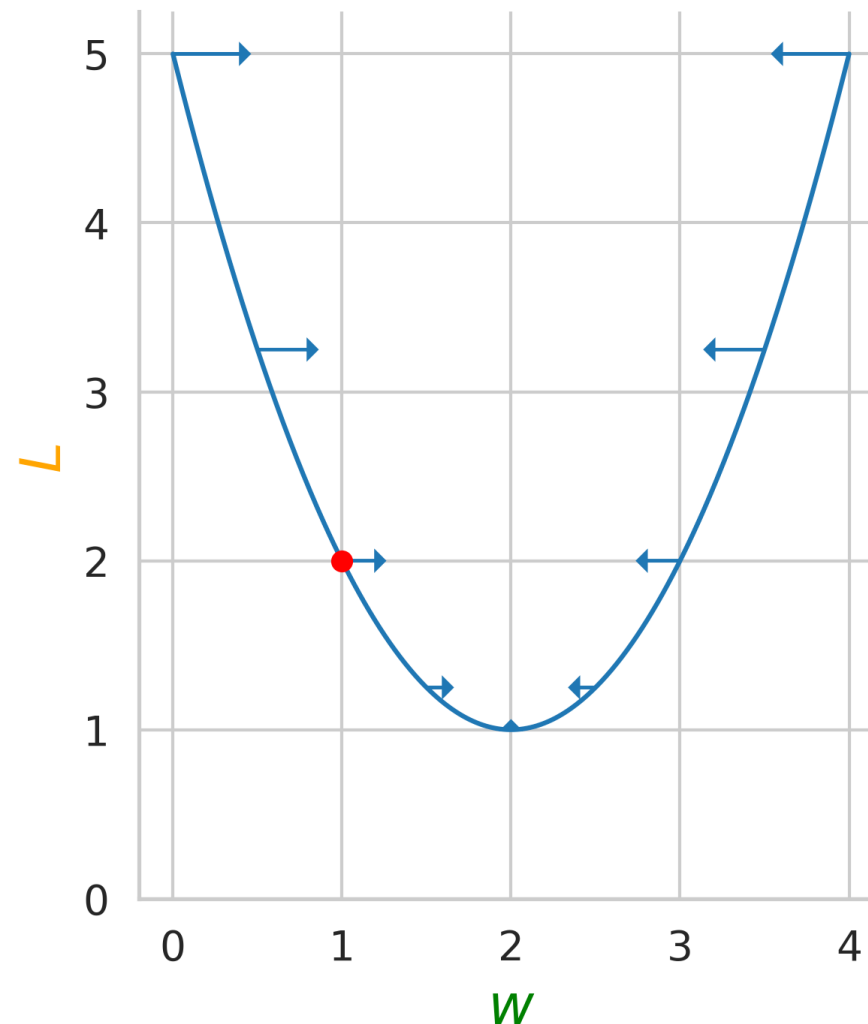
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- GRADIENT DESCENT:
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.



Gradient descent: toy example

EXAMPLE:

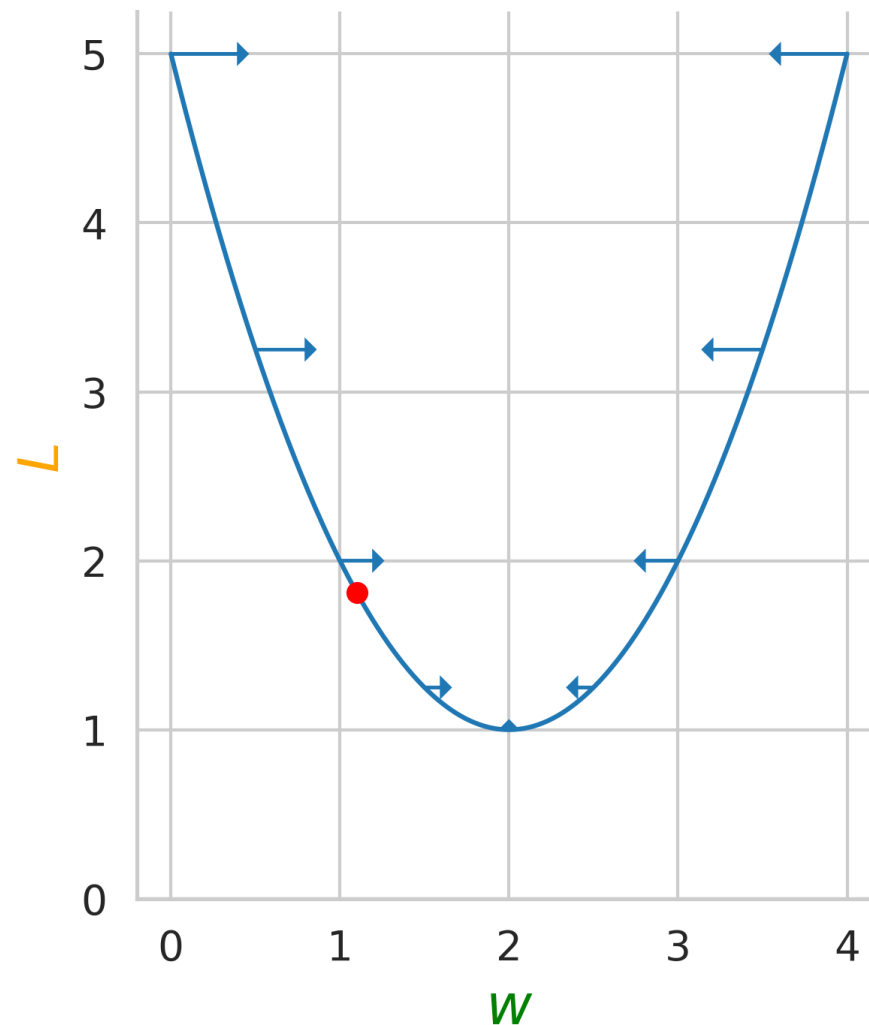
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- GRADIENT DESCENT:
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.



Gradient descent: toy example

EXAMPLE:

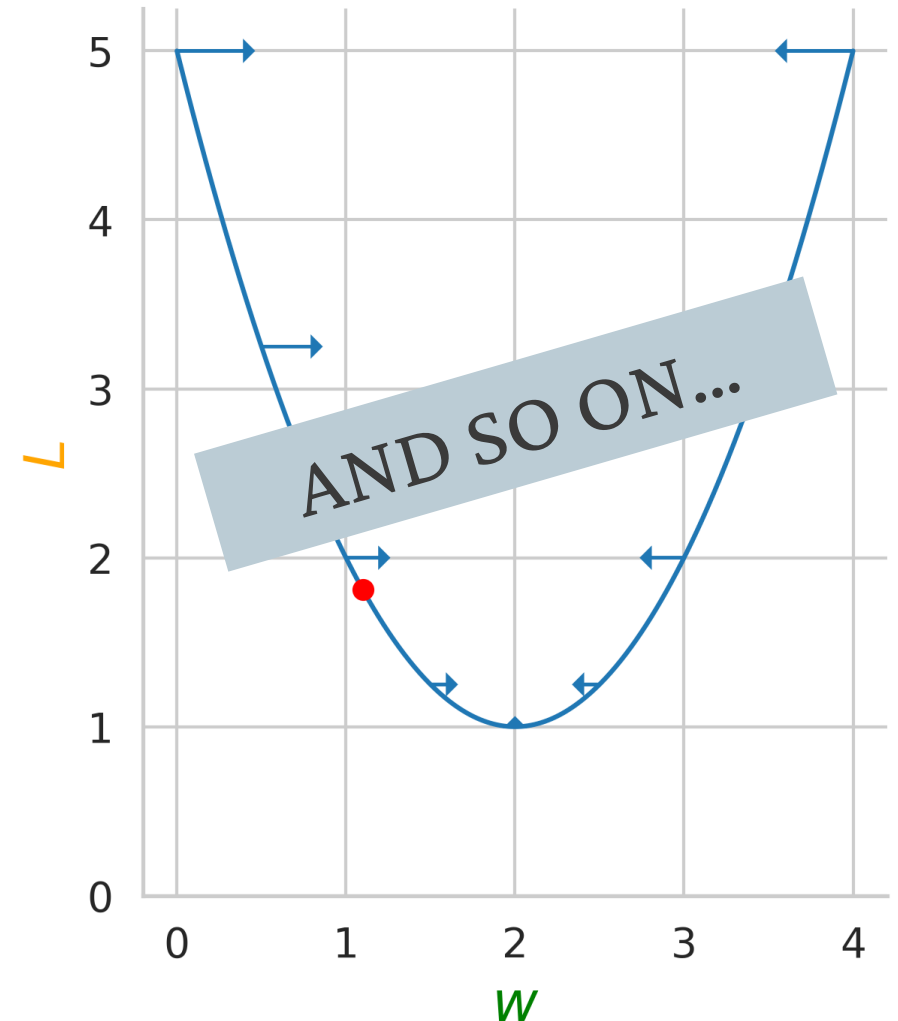
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- GRADIENT DESCENT:
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.



Gradient descent: toy example

EXAMPLE:

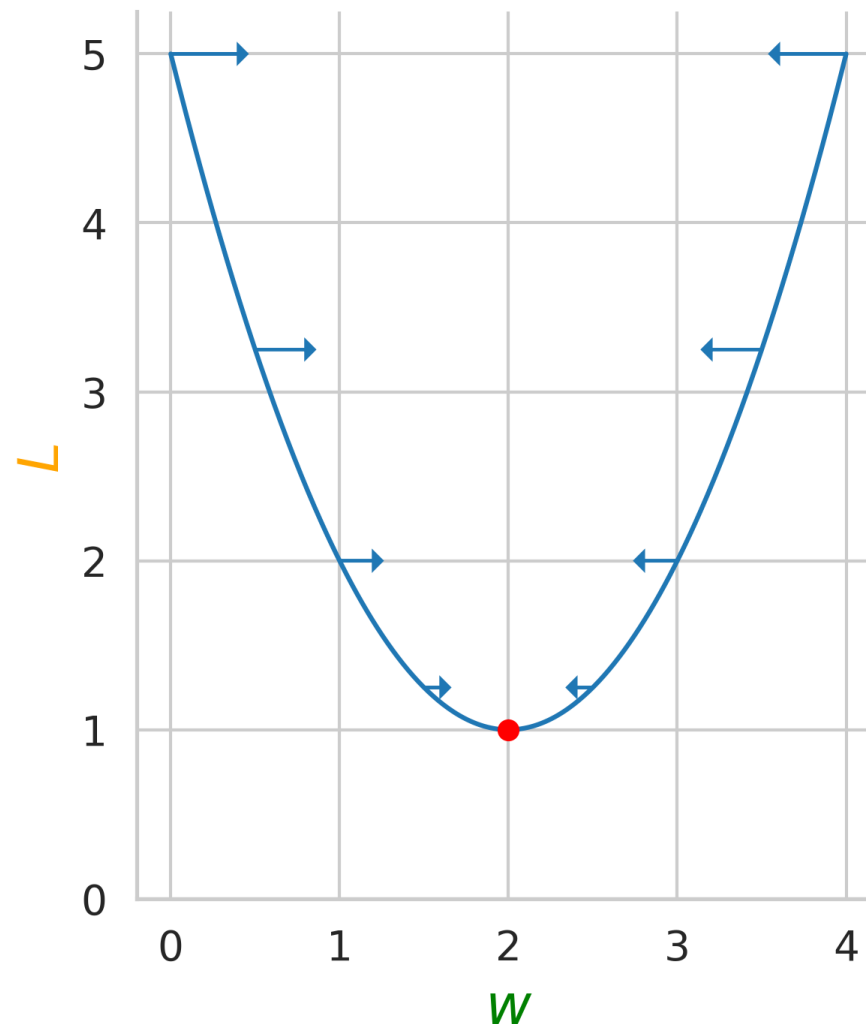
- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- GRADIENT DESCENT:
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.



Gradient descent: toy example

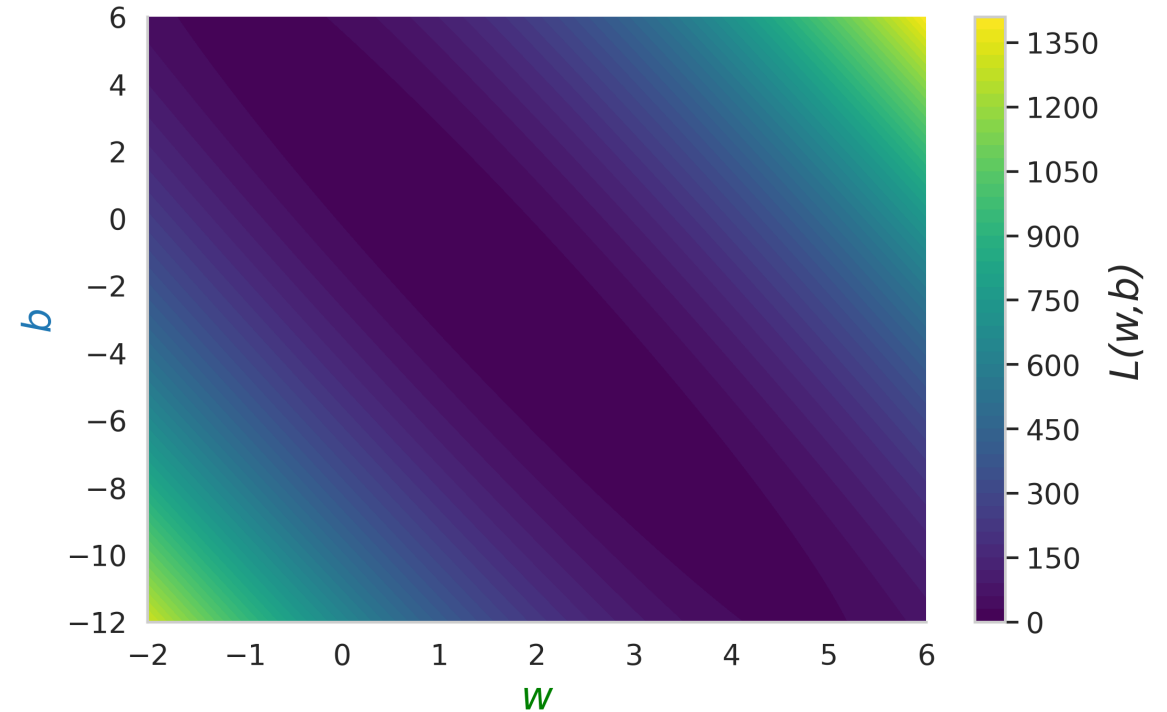
EXAMPLE:

- Want to minimize loss function $L(w)$.
- Assume only *one* input w for now.
- **GRADIENT DESCENT:**
 1. Choose w randomly.
 2. Calculate derivative/gradient $\frac{dL}{dw}$
 3. Move w a little in *opposite* direction of gradient.
 4. Repeat steps 2–3.
- *Minimum has been found!*
- What if there are *two* inputs, $L(w, b)$?



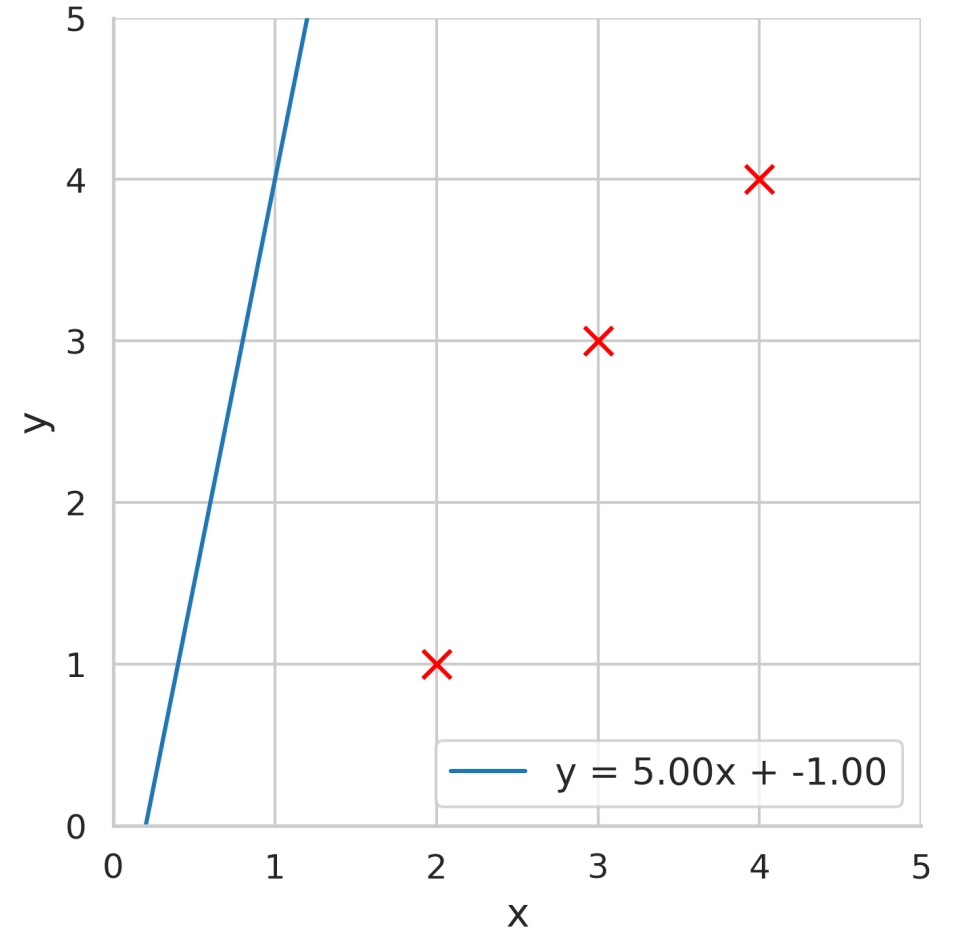
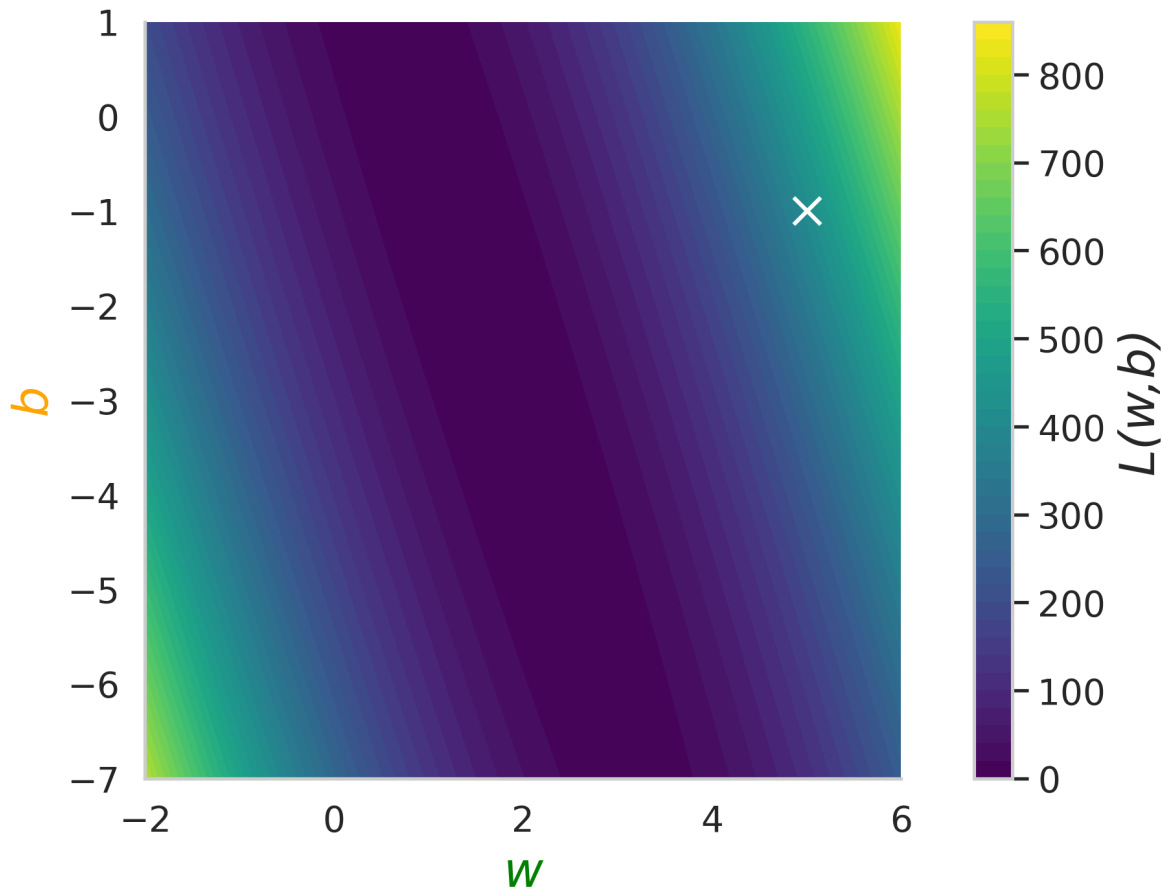
Back to linear regression

- Remember: want to find w, b that minimizes
$$L(w, b) = \sum_{n=1}^3 (wx_n + b - y_n)^2$$
- 2D optimization problem. Not a curve, but an entire loss *surface*!
- Every point represents a choice of w, b .
- Yellow \rightarrow higher loss \rightarrow worse fit.
- Blue \rightarrow lower loss \rightarrow better fit.



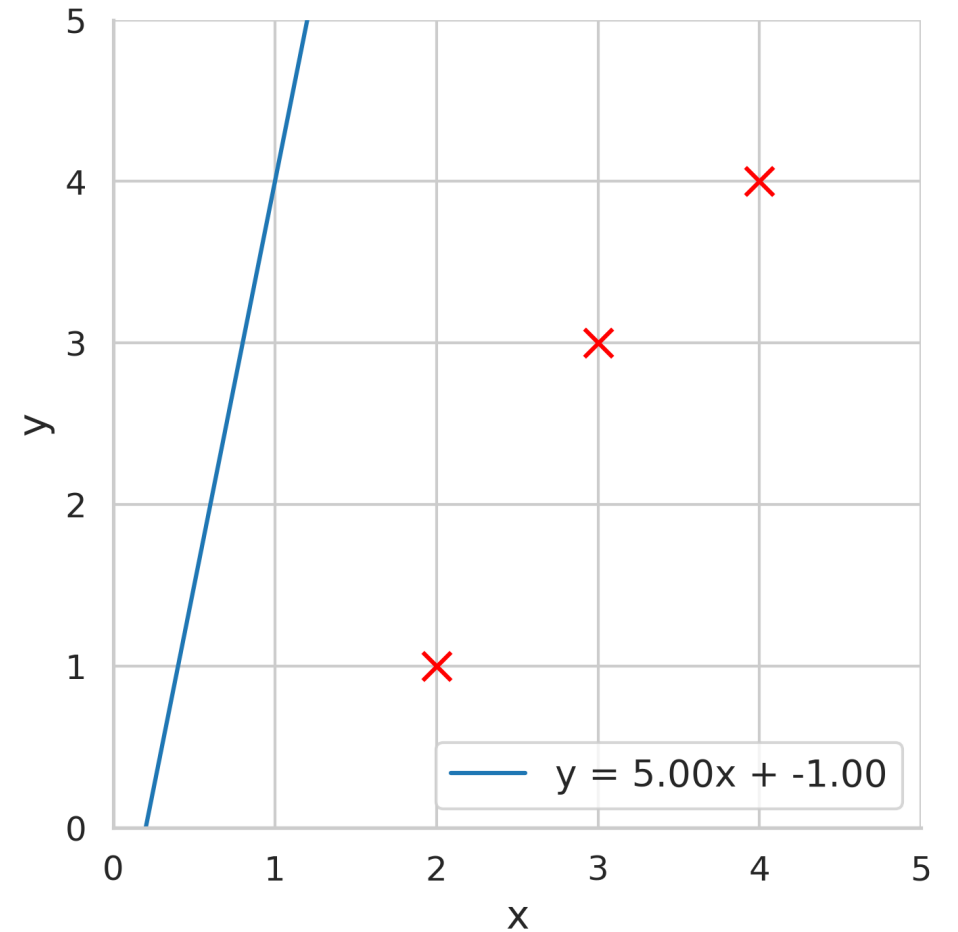
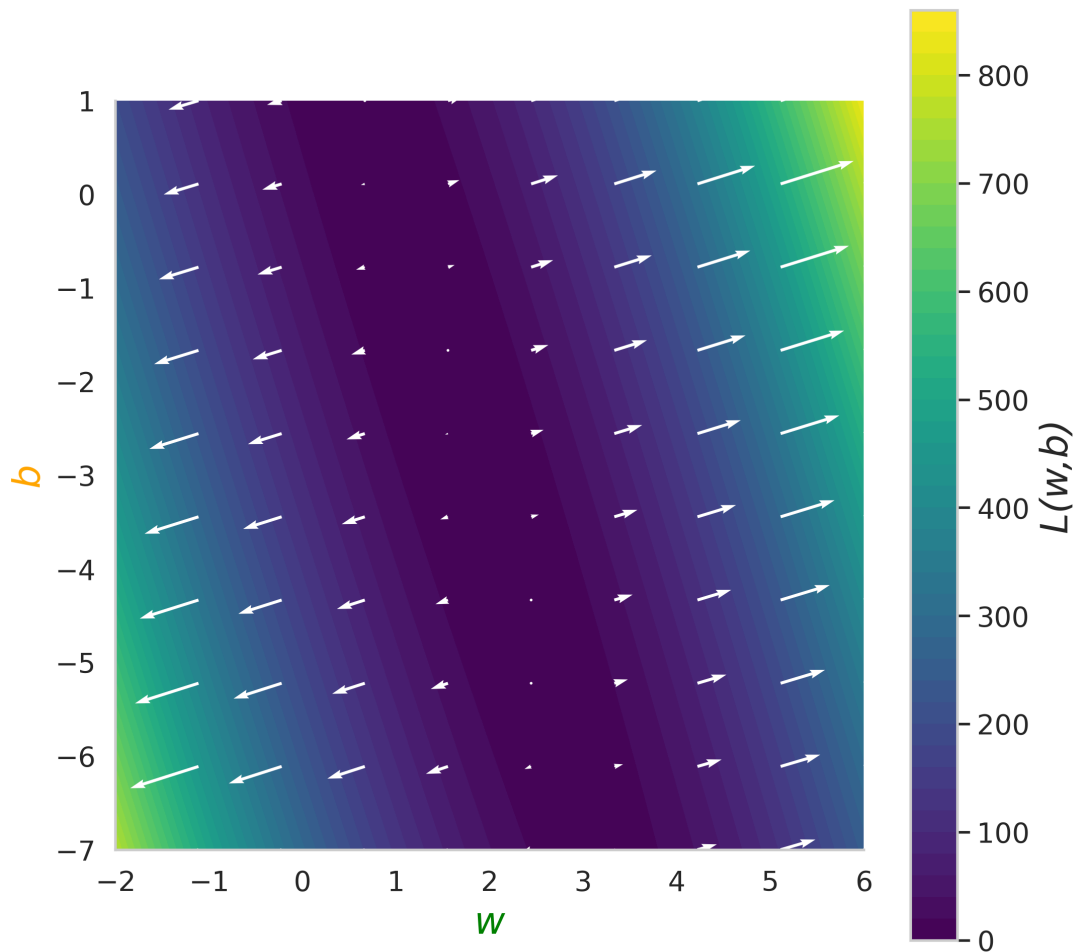
Gradient descent for linear regression

1. Choose w , b randomly (terrible fit).



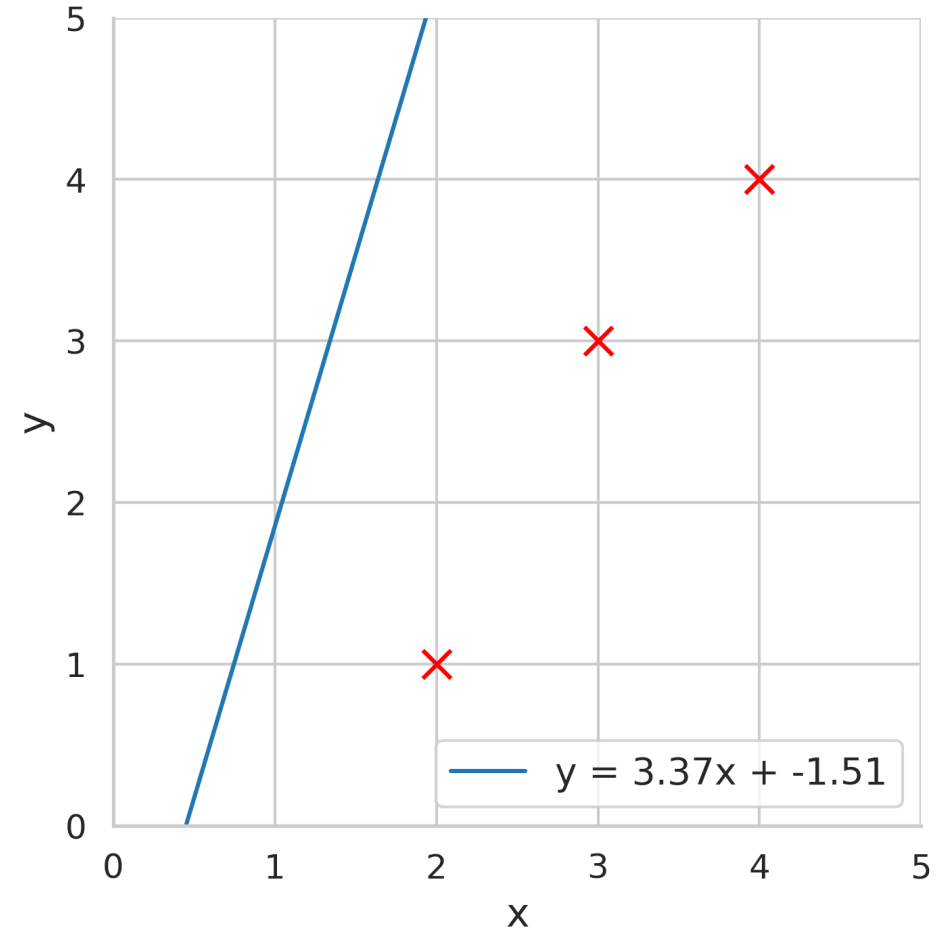
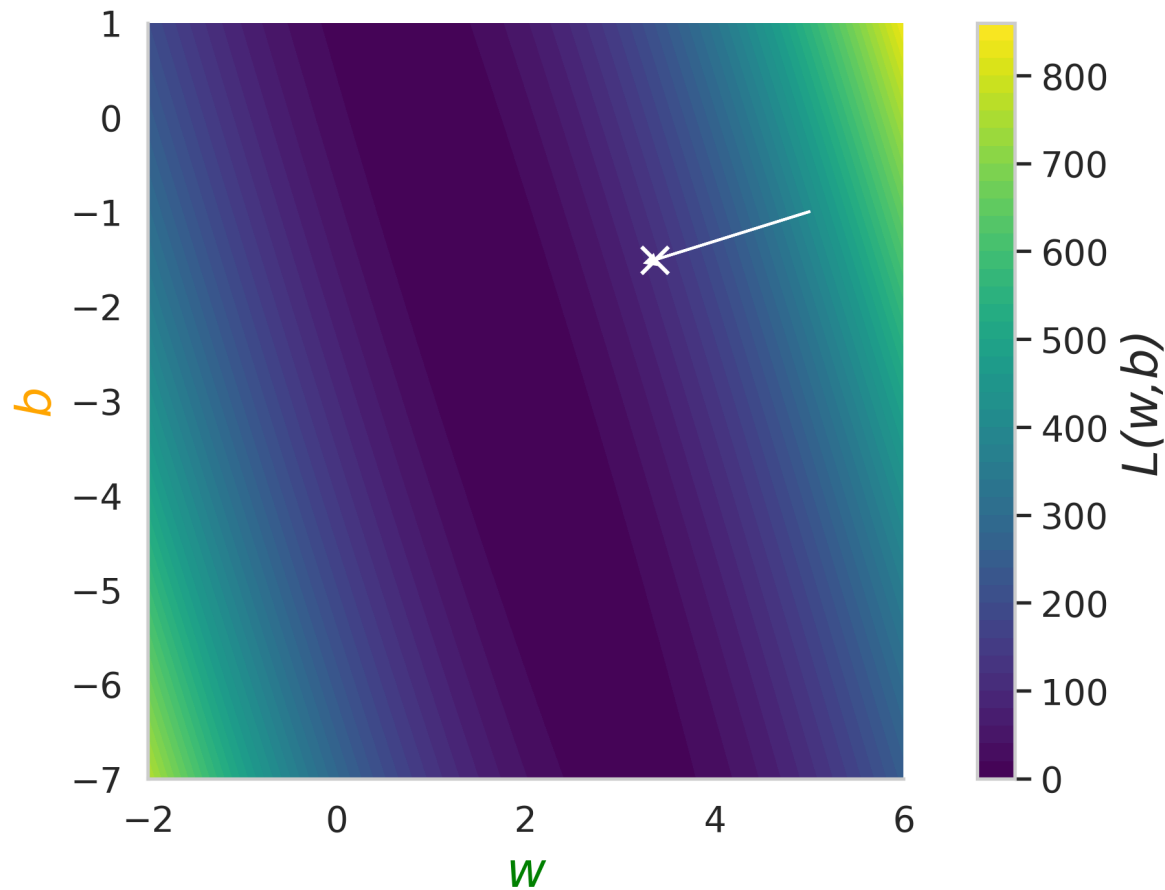
Gradient descent for linear regression

2. Calculate derivative/gradient.



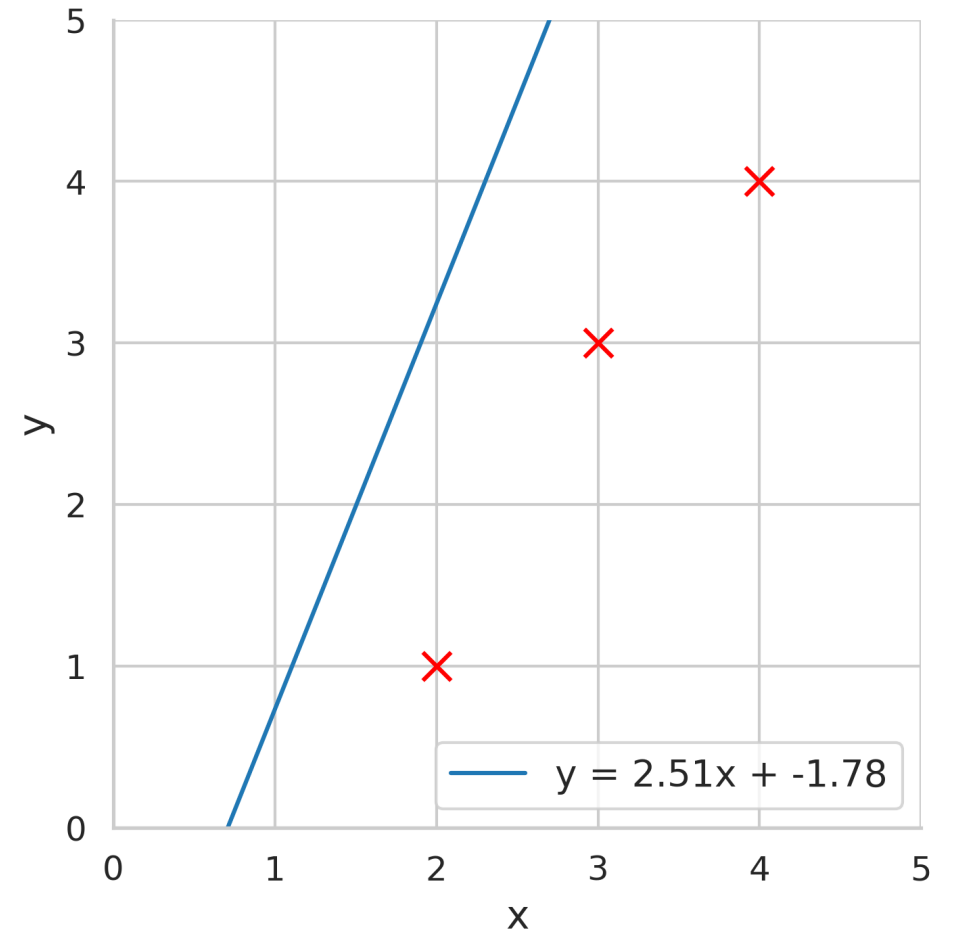
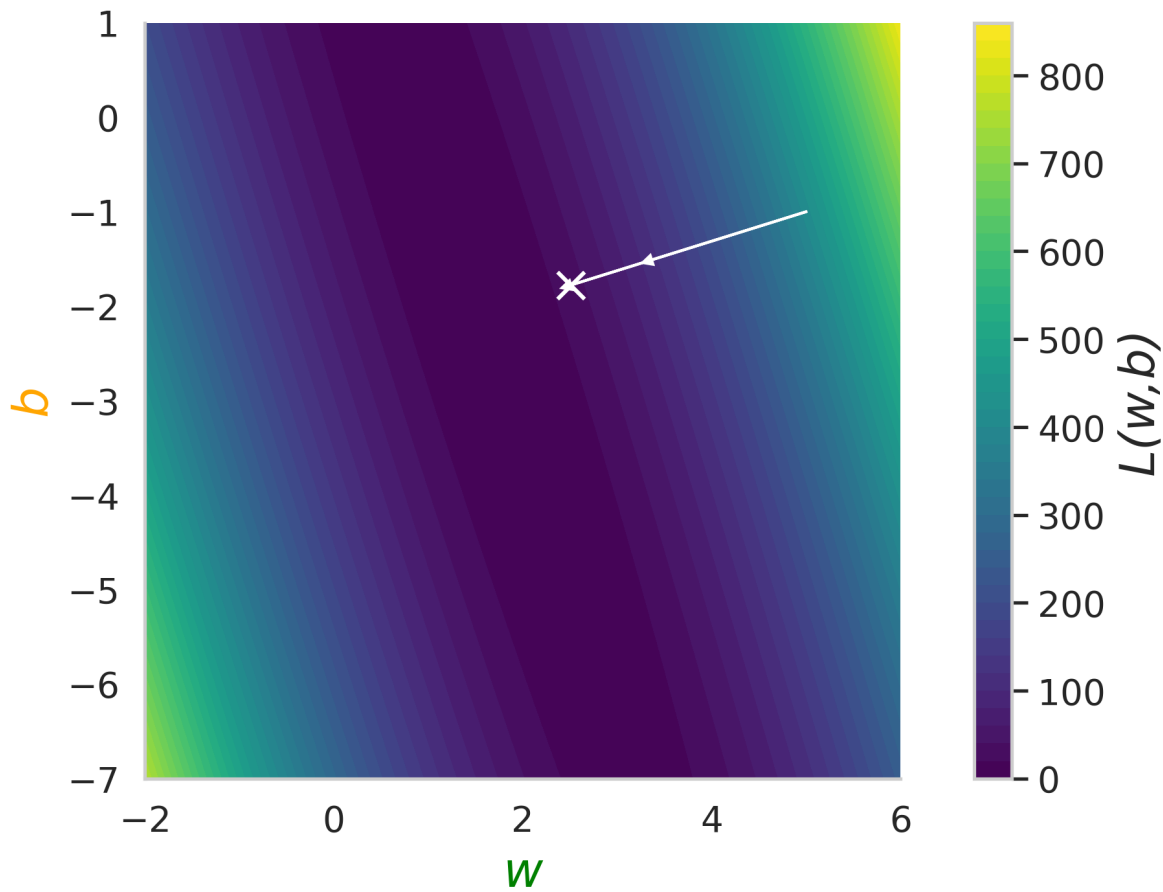
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



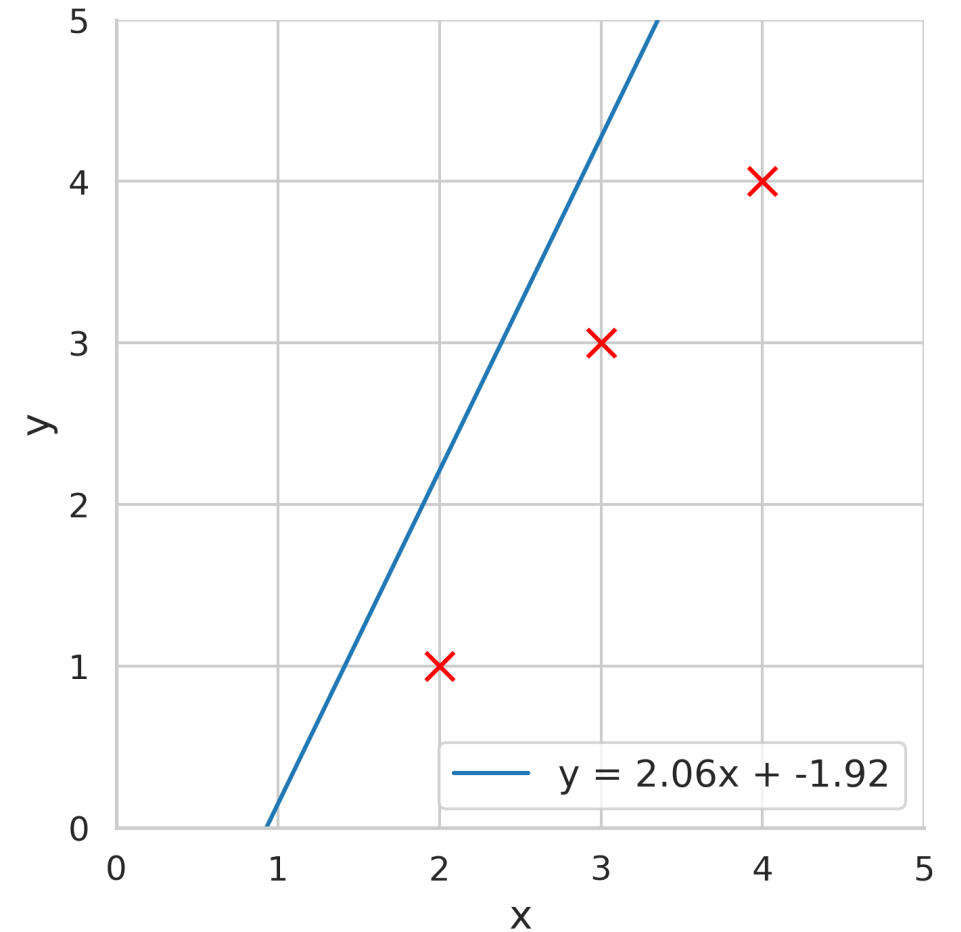
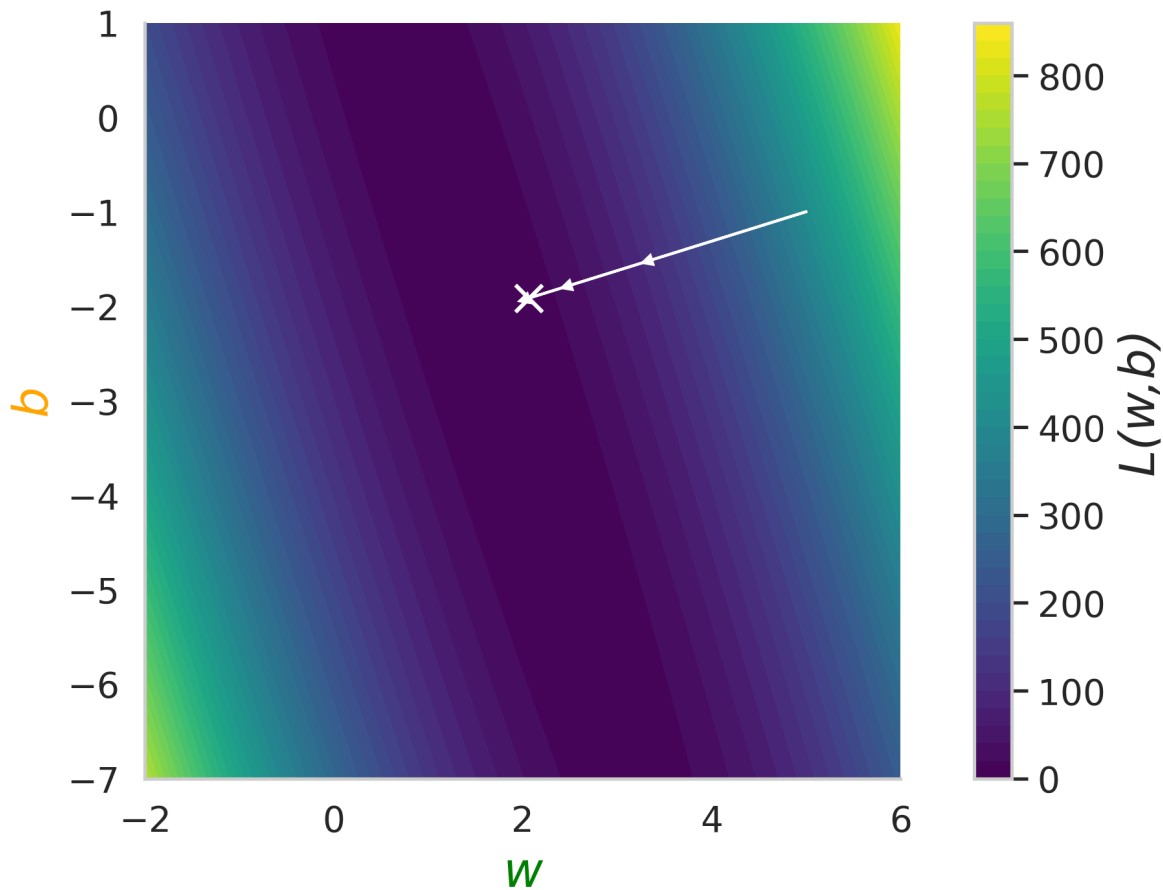
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



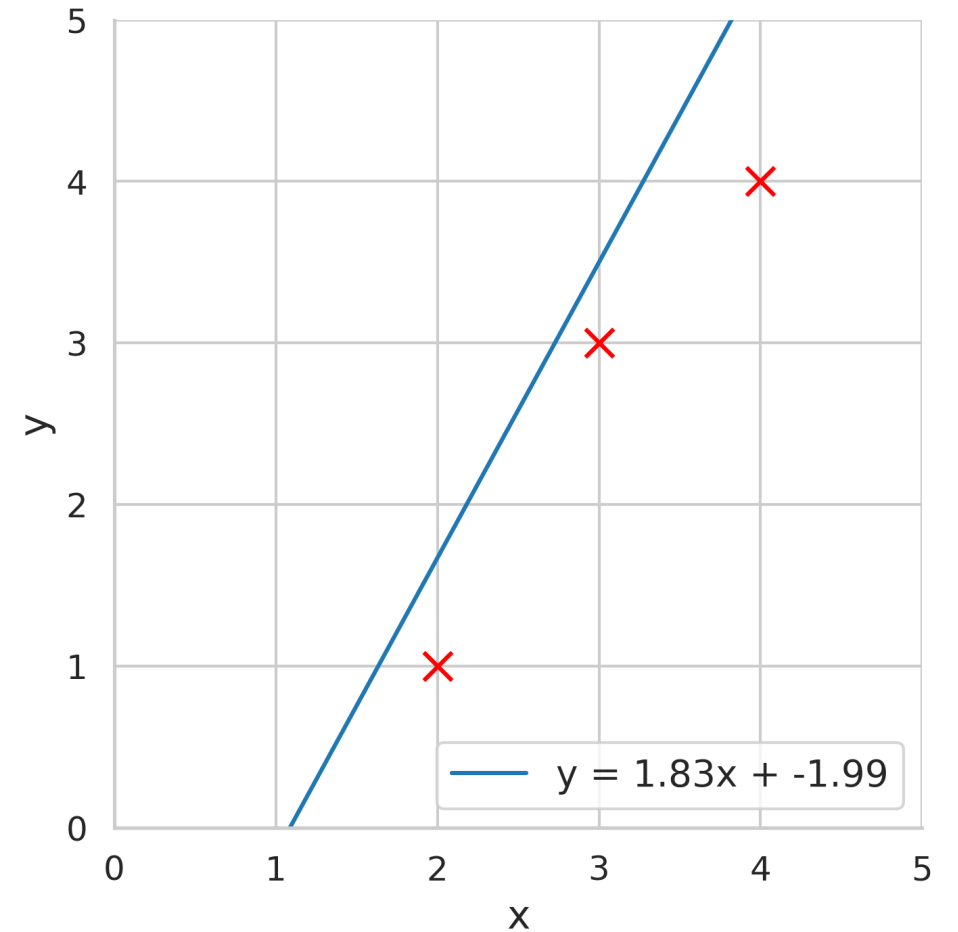
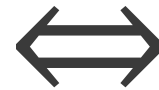
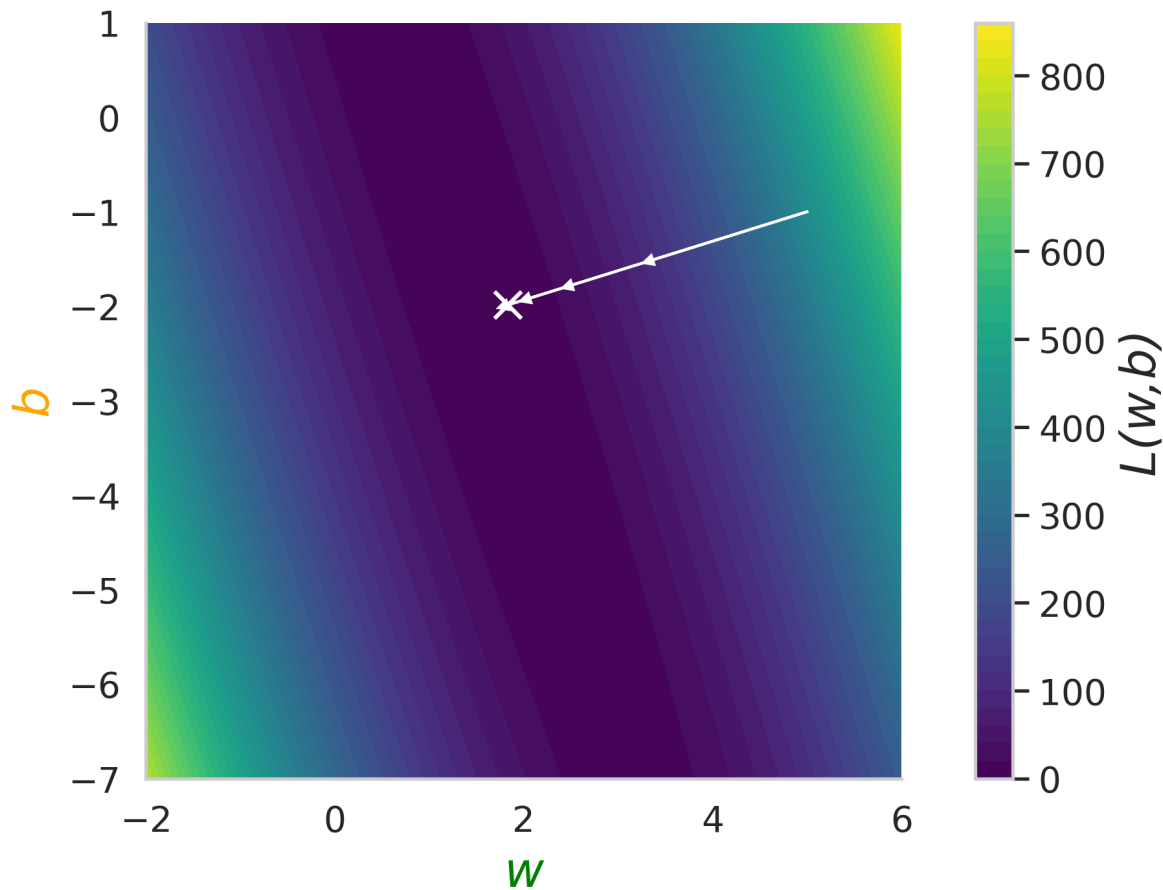
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



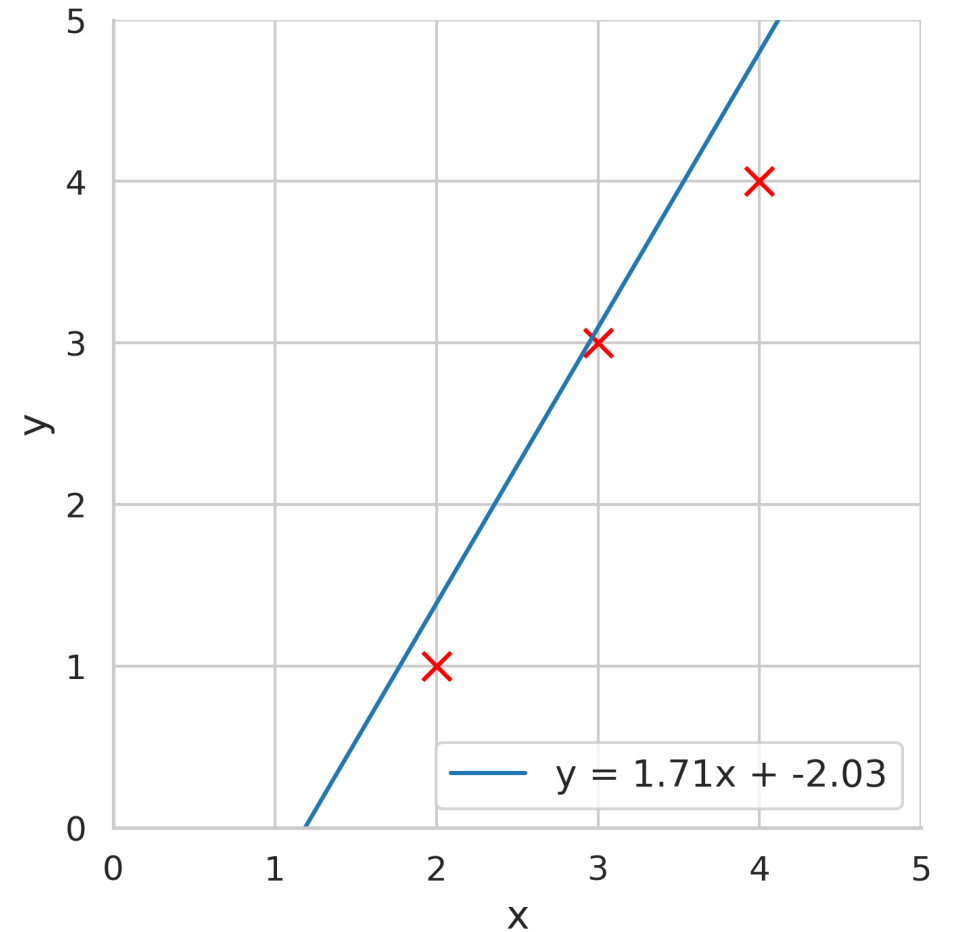
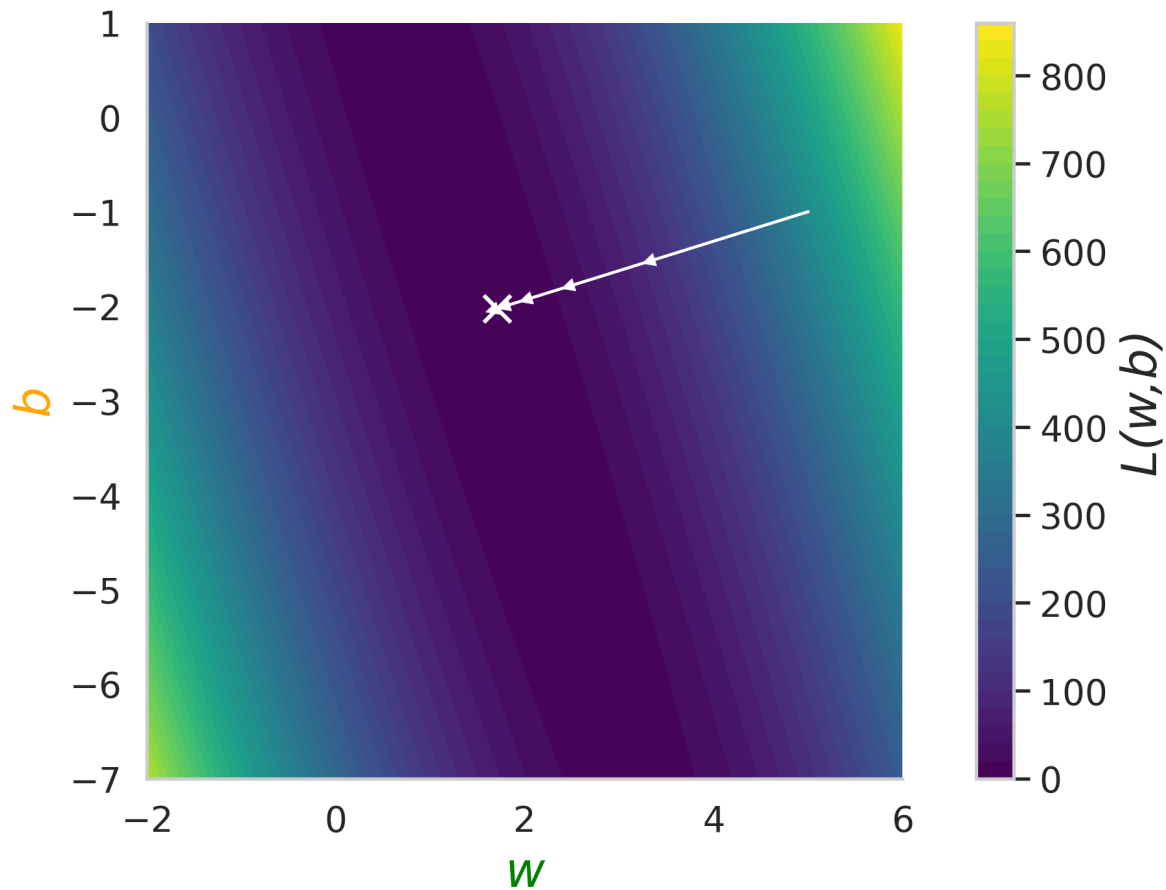
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



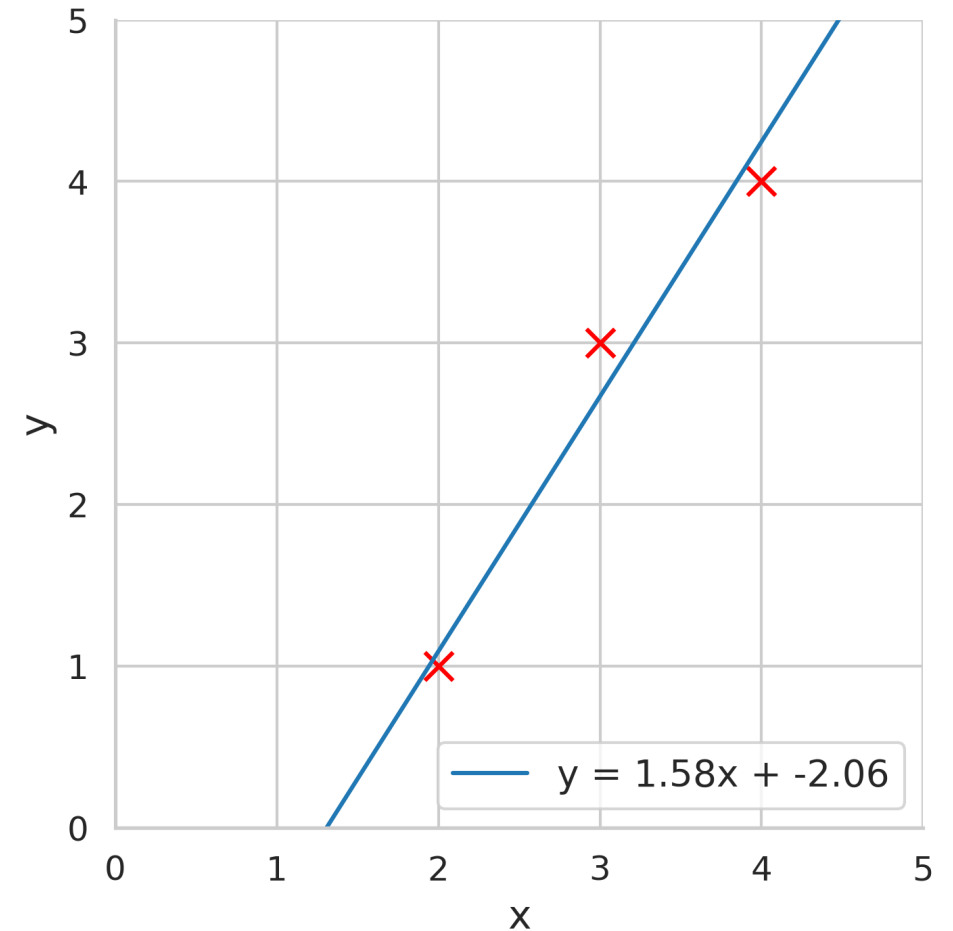
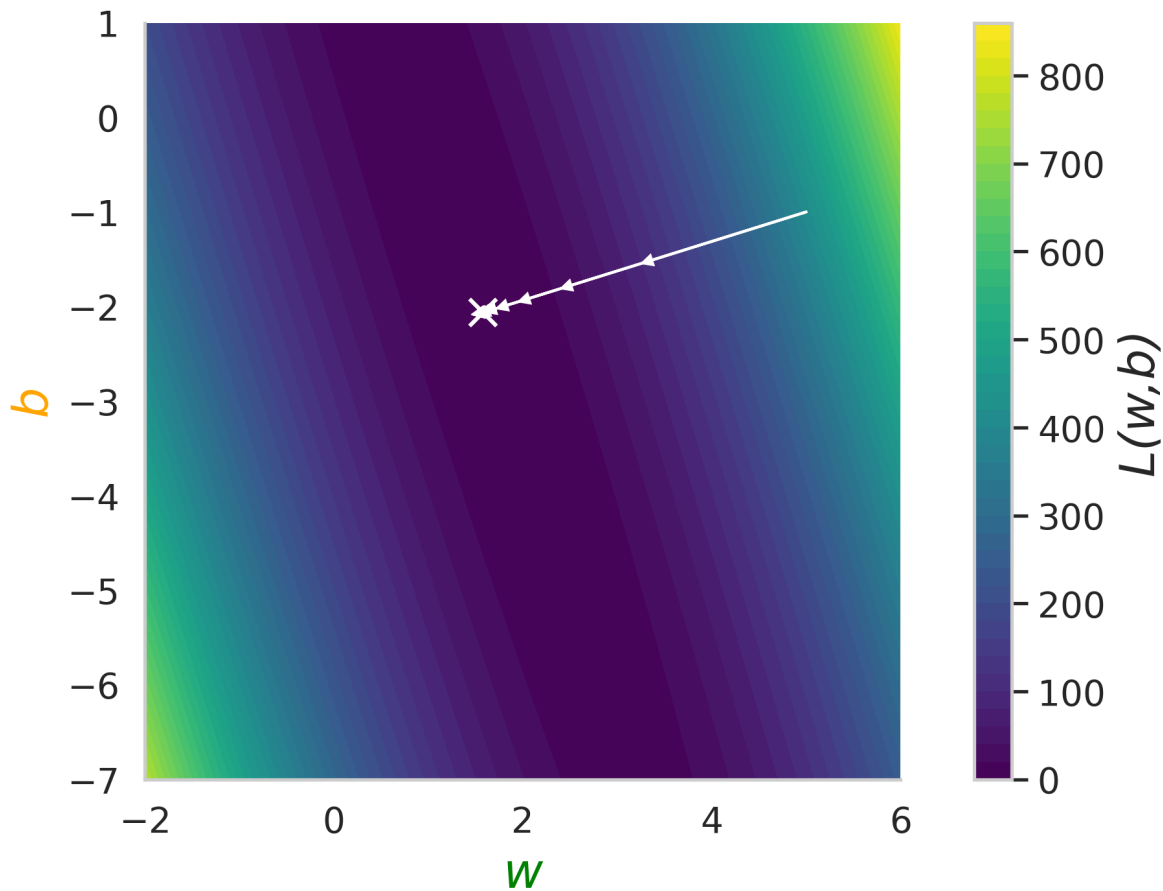
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



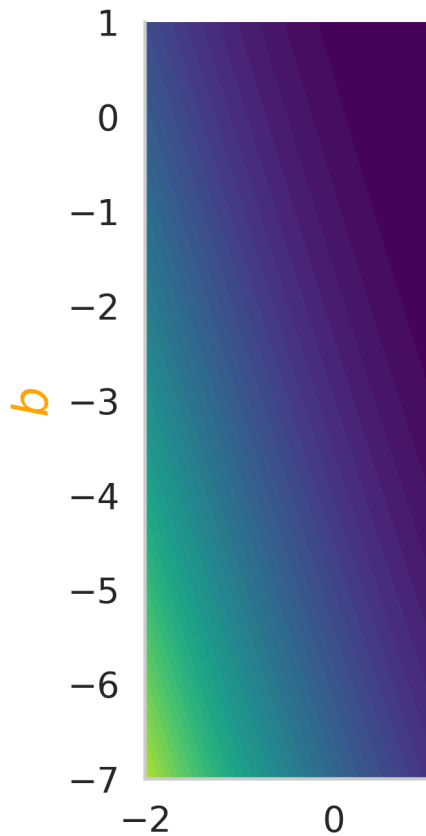
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



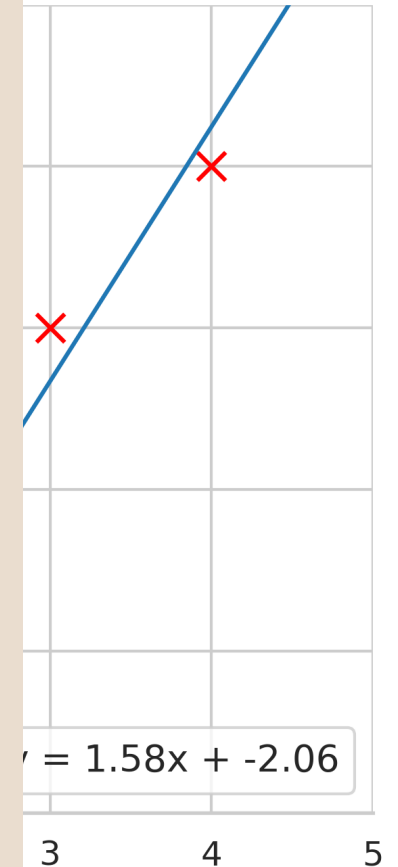
Gradient descent for linear regression

2. Calculate derivative/gradient.
3. Move w, b a little in opposite direction of gradient.



*Best-fit line found
using gradient
descent!*

Can think of this as *learning*, with
gradient descent as the *learning algorithm*



Question & Answer