Mathematical Concepts (1/2)
Functions, Minimization, Gradient

Fundamentals of Artificial Intelligence

Instructor: Chenhui Chu

Email: chu@i.kyoto-u.ac.jp

Teaching Assistant: Youyuan Lin

E-mail: youyuan@nlp.ist.i.kyoto-u.ac.jp

Schedule

- 1. Overview of AI and this Course (4/14)
- 2. Introduction to Python (4/21)
- 3, 4. Mathematics Concepts I, II (4/28, 5/12)
- 5, 6. Regression I, II (5/19, 5/26)
- 7. Classification (6/2)
- 8. Introduction to Neural Networks (6/9)
- 9. Neural Networks Architecture and Backpropagation (6/16)
- 10. Fully Connected Layers (6/23)
- 11, 12, 13. Computer Vision I, II, III (6/30, 7/7, 7/14)
- 14. Natural Language Processing (7/17)

Overview of This Course

11, 12, 13. Computer Vision I, II, III

14. Natural language processing

Deep Learning Applications

8. Neural network Introduction

9. Architecture and Backpropagation

10. Feedforward neural networks

Deep Learning

5. Simple linear regression

6. Multiple linear regression

7. Classification

Basic Supervised Machine Learning

2. Python

3, 4. Mathematics Concepts I, II

Fundamental of Machine Learning

What We Are Going to Study/Review (1/3)

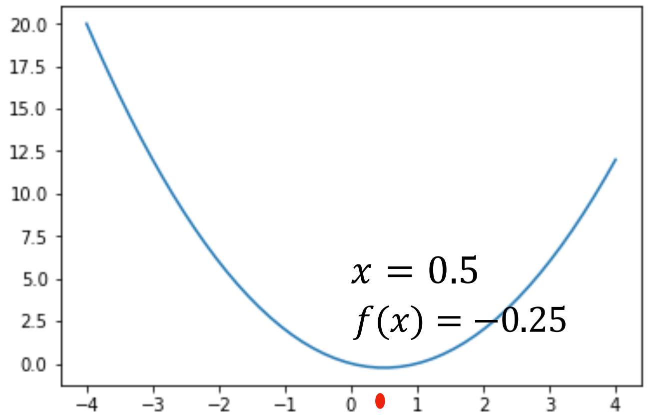
- Functions of one variable
- Functions of several variables
- Derivatives and gradient
- Finding the minimum of a function with gradient descent

What We Are Going to Study/Review (2/3)

- Given a function of **one variable**, find practically the value for which it is minimum
  - a.k.a "univariate function"
  - You should have seen how to do that for simple functions in high school

$$f: \mathbb{R} \rightarrow \mathbb{R}$$
$$f(x) = x^2 - x$$

$$\underset{x}{\operatorname{argmin}} f(x)$$



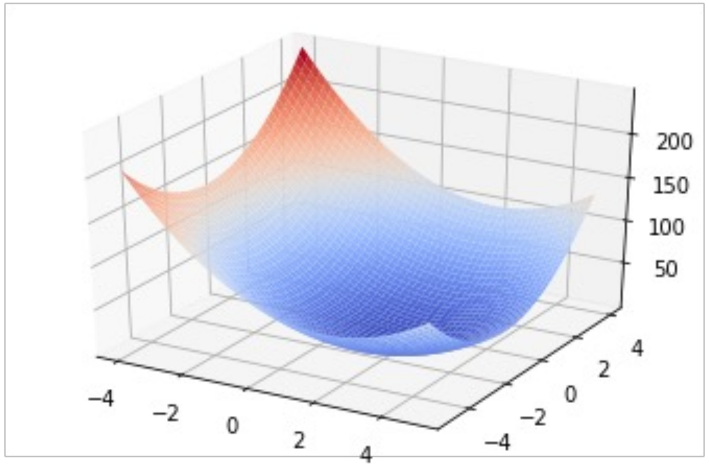
What We Are Going to Study/Review (3/3)

- Given a function of **several variables**, find the value for which it is minimum
  - a.k.a "multivariate function"

$$f: \mathbb{R}^2 \to \mathbb{R}$$

$$f(x, y) = (x + y)^2 + 1$$

$$\underset{x,y}{\operatorname{argmin}} f(x, y)$$



A Peek in the Future

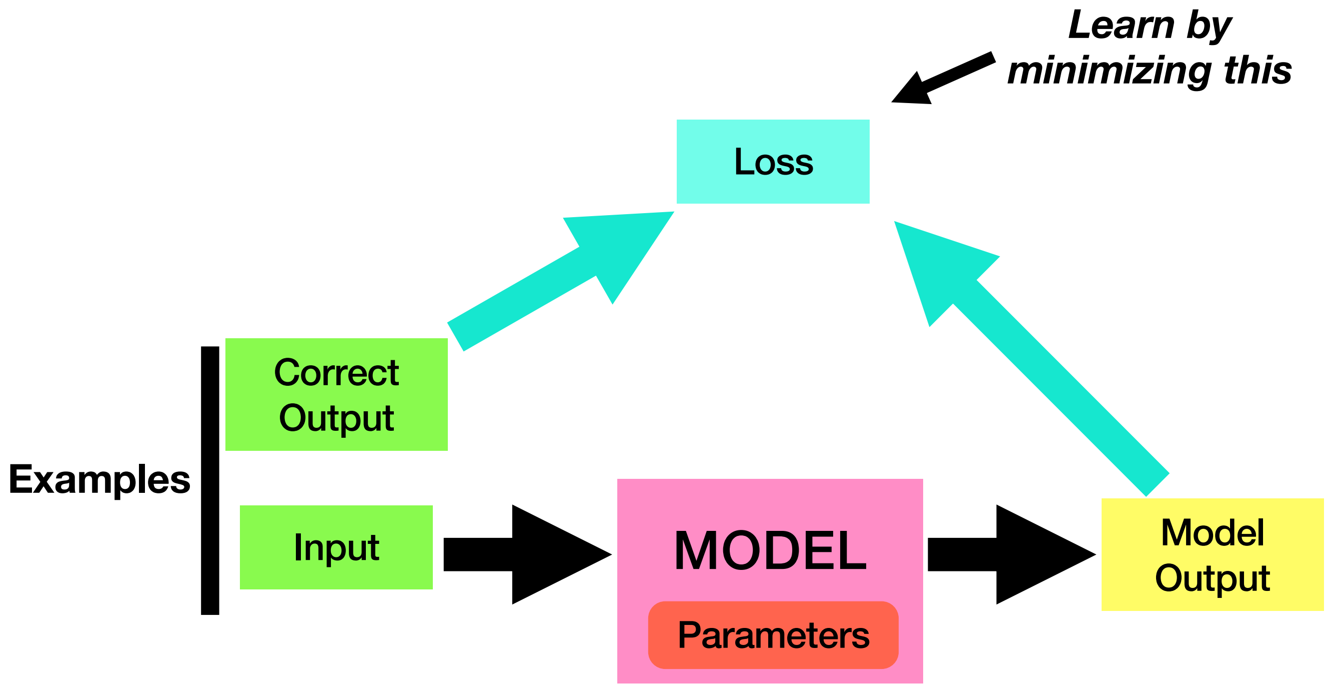
Why We Do it?

- Actually, almost all algorithms of supervised machine learning consist in finding the **minimum** of a **function of several variables**

A Peek in the Future

Supervised Learning (1/6)

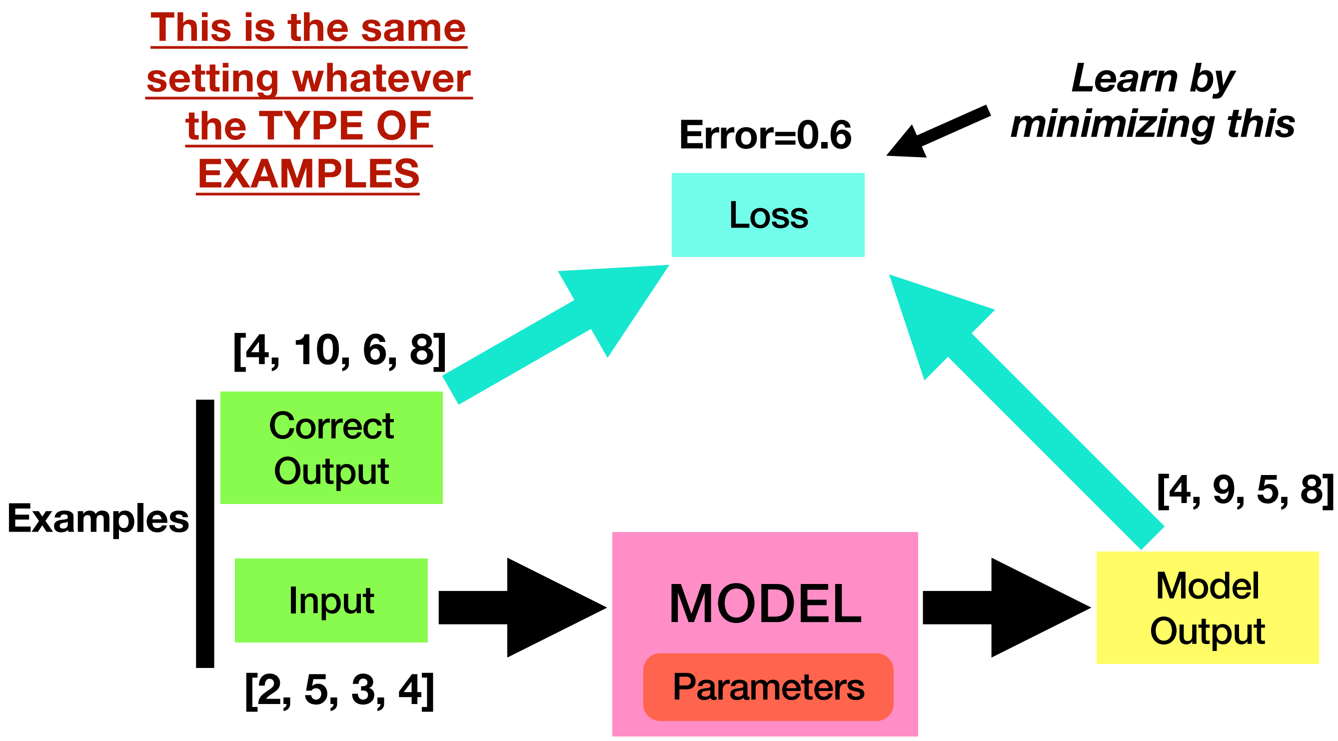
- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



A Peek in the Future

Supervised Learning (2/6)

- In supervised learning, we usually have:
  - A **MODEL**: a "parameterized" function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



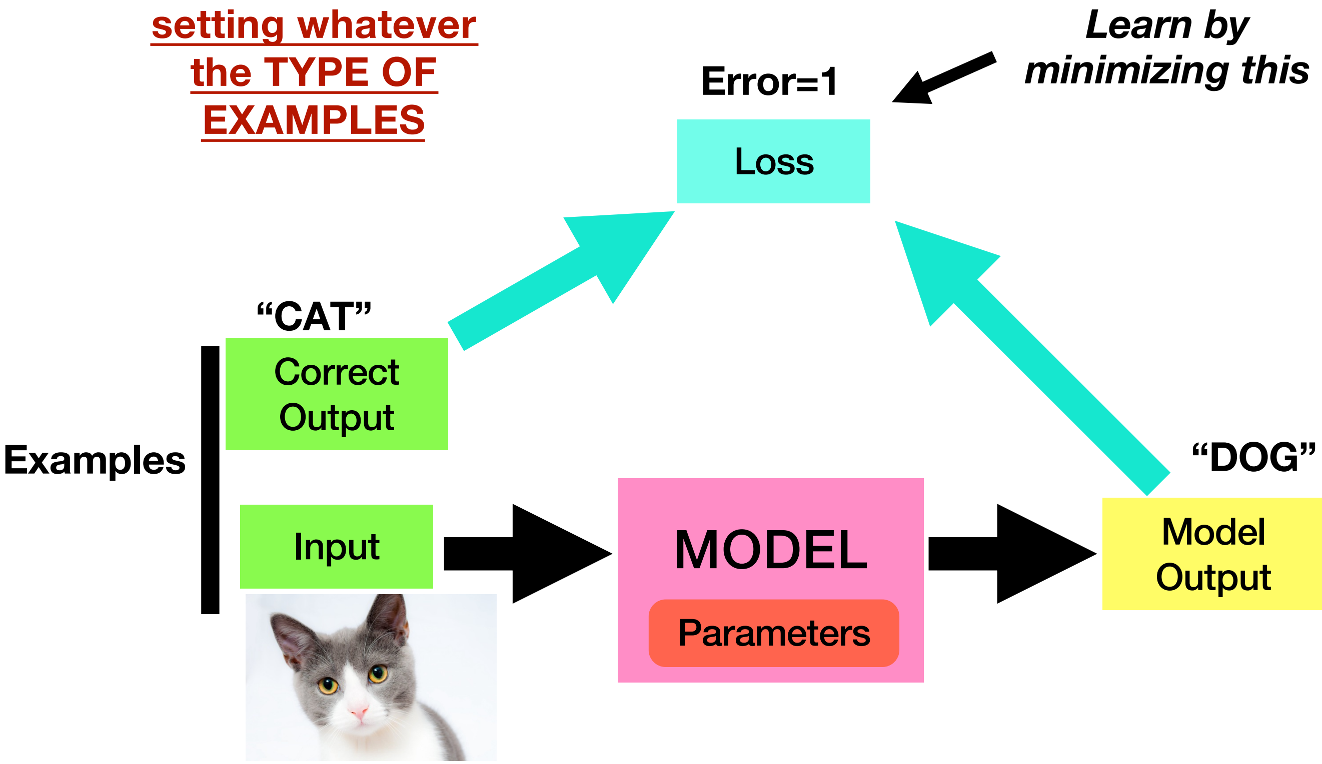
Learning to multiply numbers by two

A Peek in the Future

Supervised Learning (3/6)

- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output

This is the same setting whatever the TYPE OF EXAMPLES

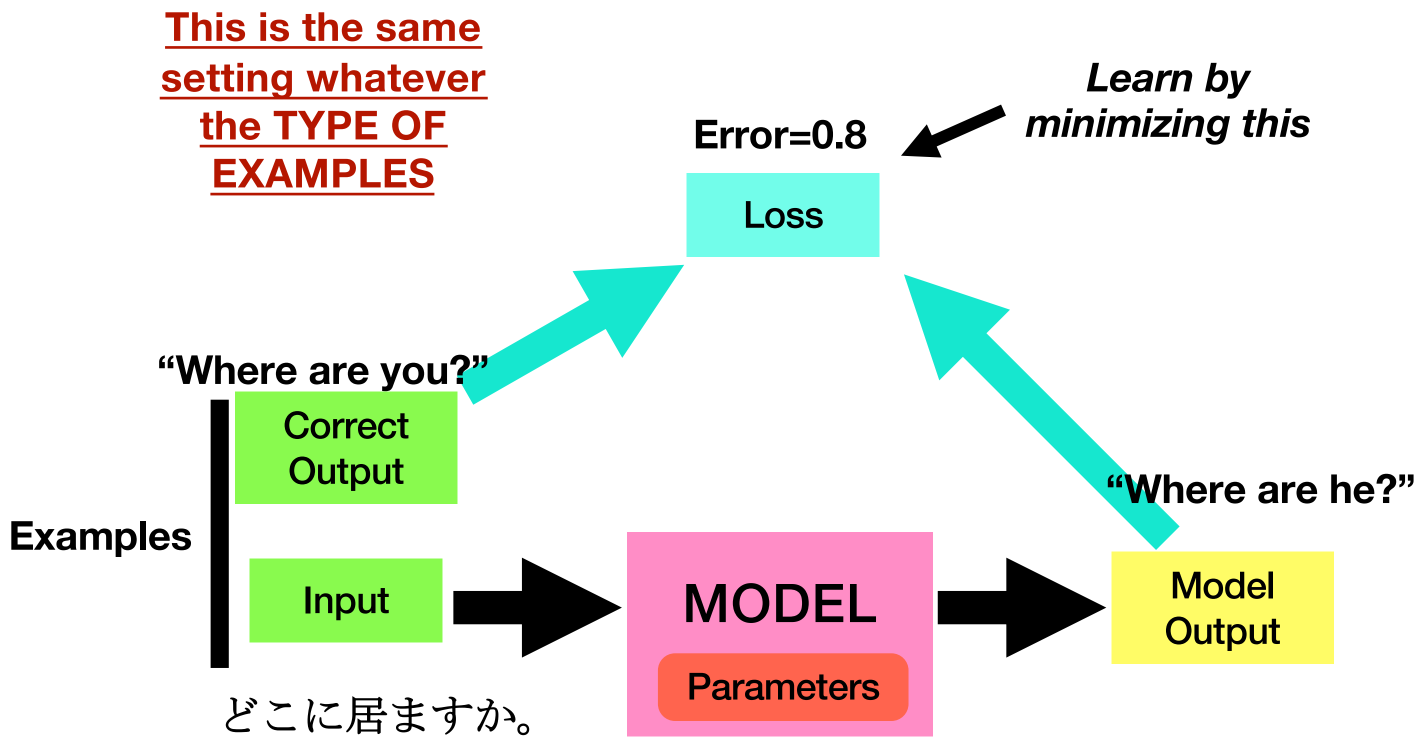


Learning to recognize images

A Peek in the Future

Supervised Learning (4/6)

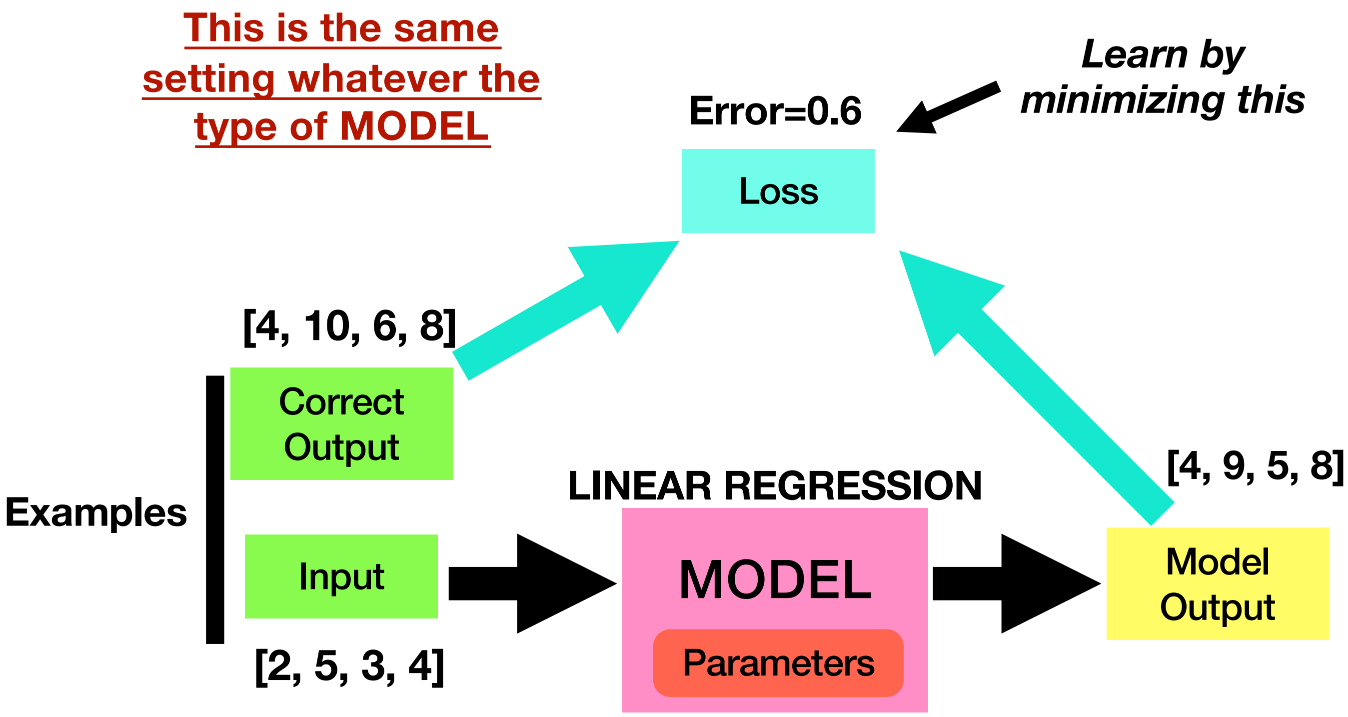
- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



Learning to translate

Supervised Learning (5/6)

- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output

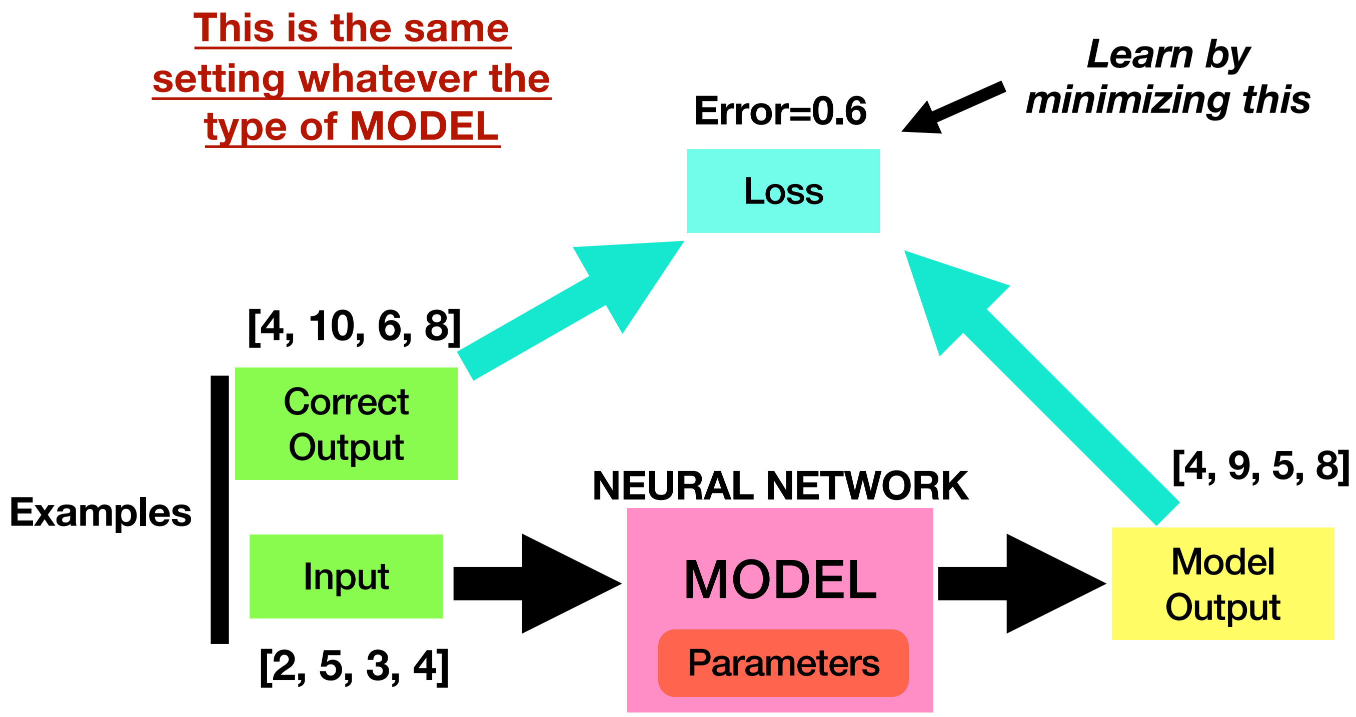


Learning to multiply numbers by two with a Linear Regression Model

A Peek in the Future

Supervised Learning (6/6)

- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that computes how different the model output is from the correct output
  - ***Examples*** of input and correct output



Learning to multiply numbers by two with a Neural Network

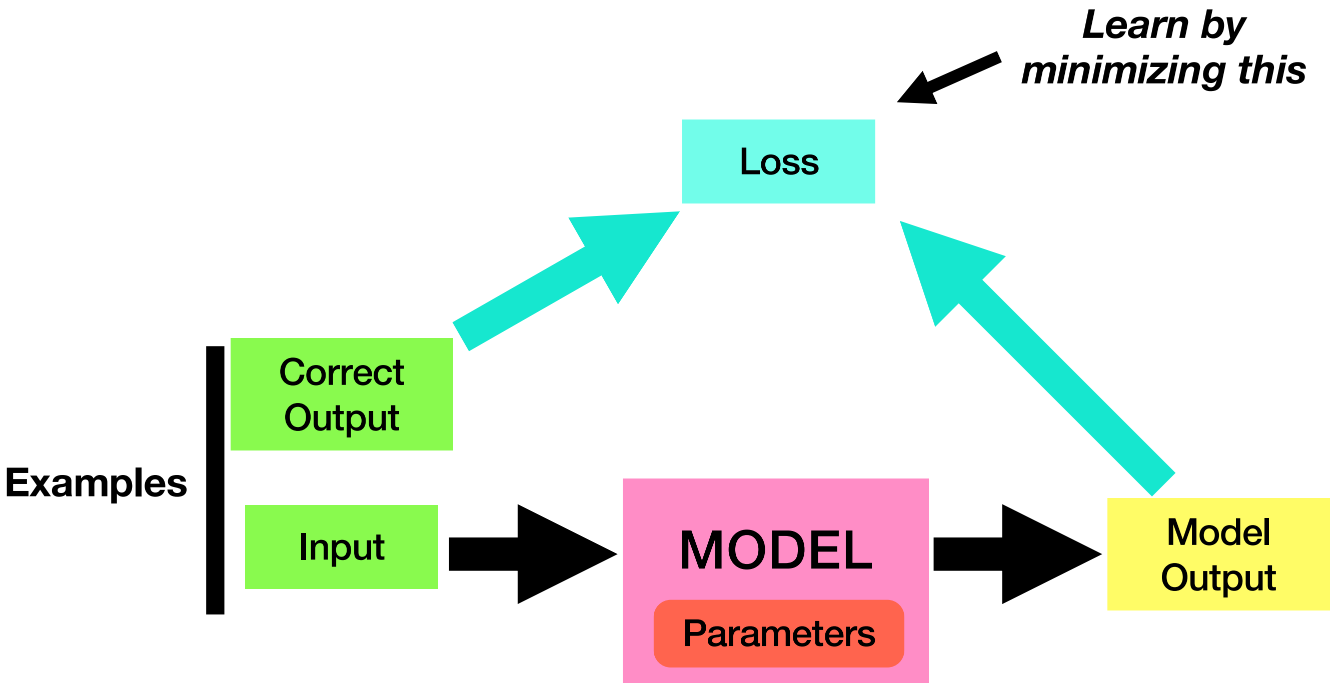
Terminology

- Because minimizing a loss is the main way for “learning,” for us, the following expressions have all the same meaning:
  - **Minimizing** the Loss of a Model for some examples
  - **Training** a Model on some examples
  - Having a Model **learn** from some examples

Supervised Learning

- We will go back to these concepts later in the semester

- For now, let us focus on methods for minimizing a function



Minimizing a Function of One Variable

Functions of One Variable

- Hopefully, you are all familiar with the concept of "*functions of one variable*"
    - Terminology: also called "*Univariate function*"
- Take a single number as input, give a single number as output

$$f : \mathbb{R} \to \mathbb{R}$$
$$f(x) = x^2 - x$$

$$f(-1) = 2$$
$$f(0) = 0$$
$$f(0.1) = -0.09$$

Minimizing a Function of One Variable

- Given a function of one variable ***f(x)***, what is the input number ***x*** that gives the smallest output number?
- We note this number $\underset{x}{\operatorname{argmin}} f(x)$
- What is $\underset{x}{\operatorname{argmin}} f(x)$ for $f(x) = x^2 + 3$?
- What is $\underset{x}{\operatorname{argmin}} f(x)$ for $f(x) = x$?
- What is $\underset{x}{\operatorname{argmin}} f(x)$ for $f(x) = x^2 - x$?

The “High School” View of Minimization (1/3)

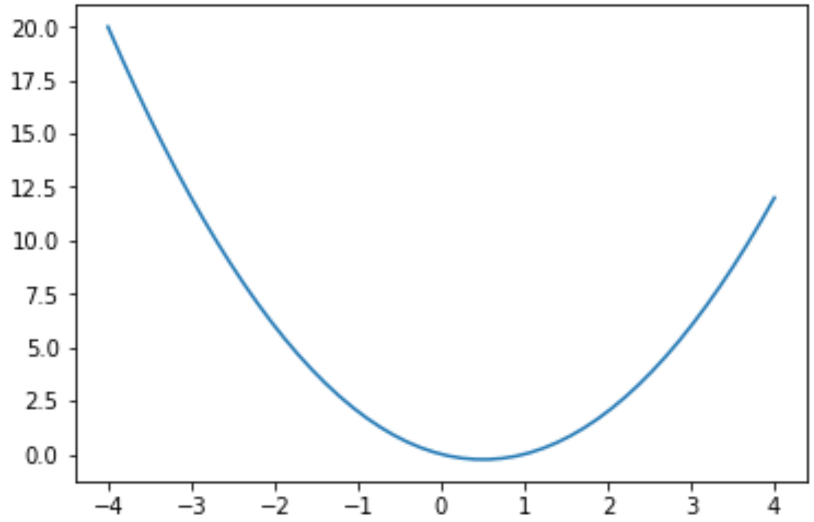
- Let us start by recalling what we learn in high school

The "High School" View of Minimization (2/3)

- Let us start by recalling what we learn in high school

To minimize f(x):

1. Compute first derivative f'(x)
2. Compute second derivative f''(x)
3. Find x0 such that f'(x0) = 0
4. If f''(x0) > 0 then x0 is a local minimum of f(x)

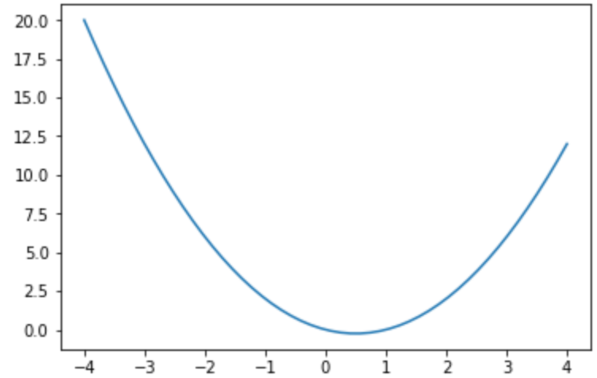


$$f: \mathbb{R} \rightarrow \mathbb{R}$$
$$f(x) = x^2 - x$$

Note: It is not how we will minimize functions in practice

The "High School" View of Minimization (3/3)

- Let us start by recalling what we learn in high school



To minimize f(x):

1. Compute first derivative f'(x)
2. Compute second derivative f''(x)
3. Find x0 such that f'(x0) = 0
4. If f''(x0) > 0 then x0 is a local minimum of f(x)

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

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

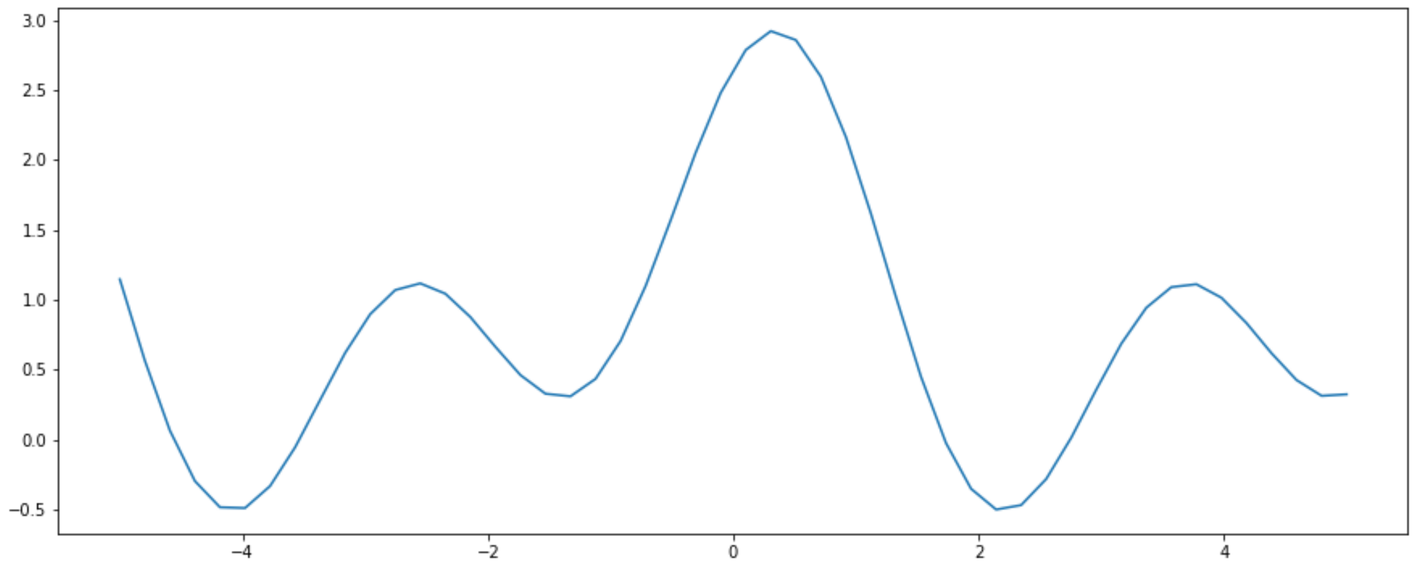
$$f'(x) = 2x - 1 \Rightarrow x_0 = 0.5$$

$$f''(x) = 2$$

Note: It is not how we will minimize functions in practice

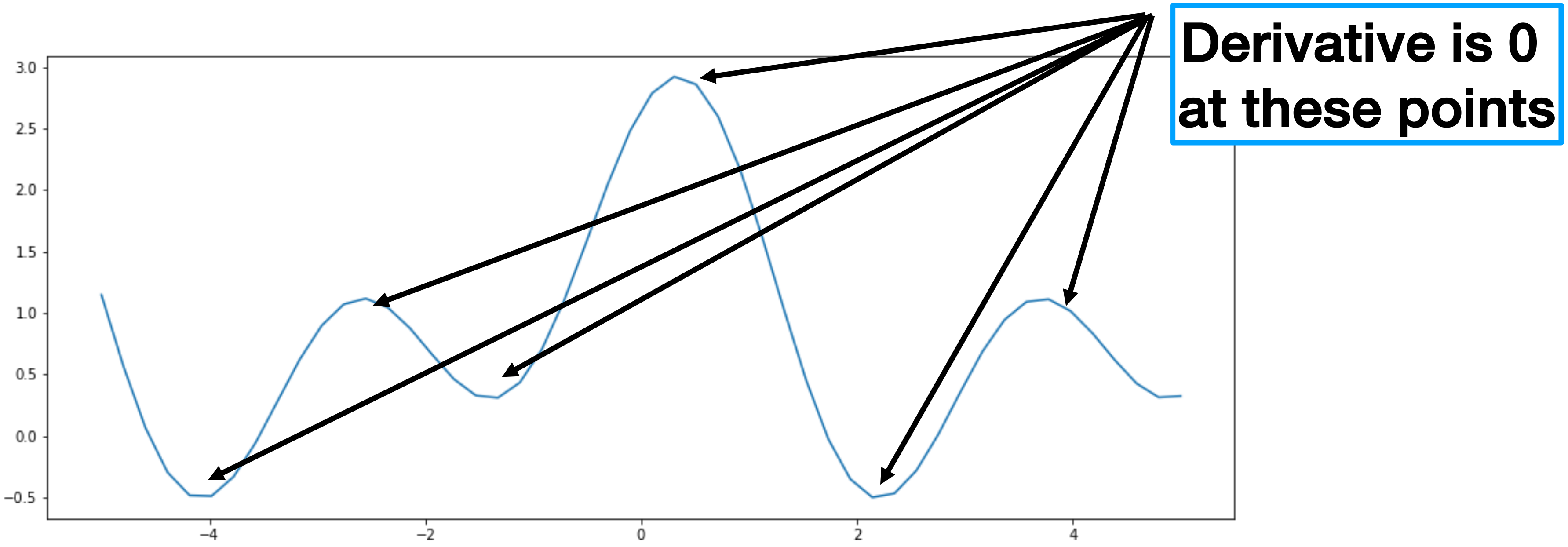
Local Minimum, Local Maximum (1/3)

- Note that the condition on the second derivative is important to distinguish **minimums** from **maximum**



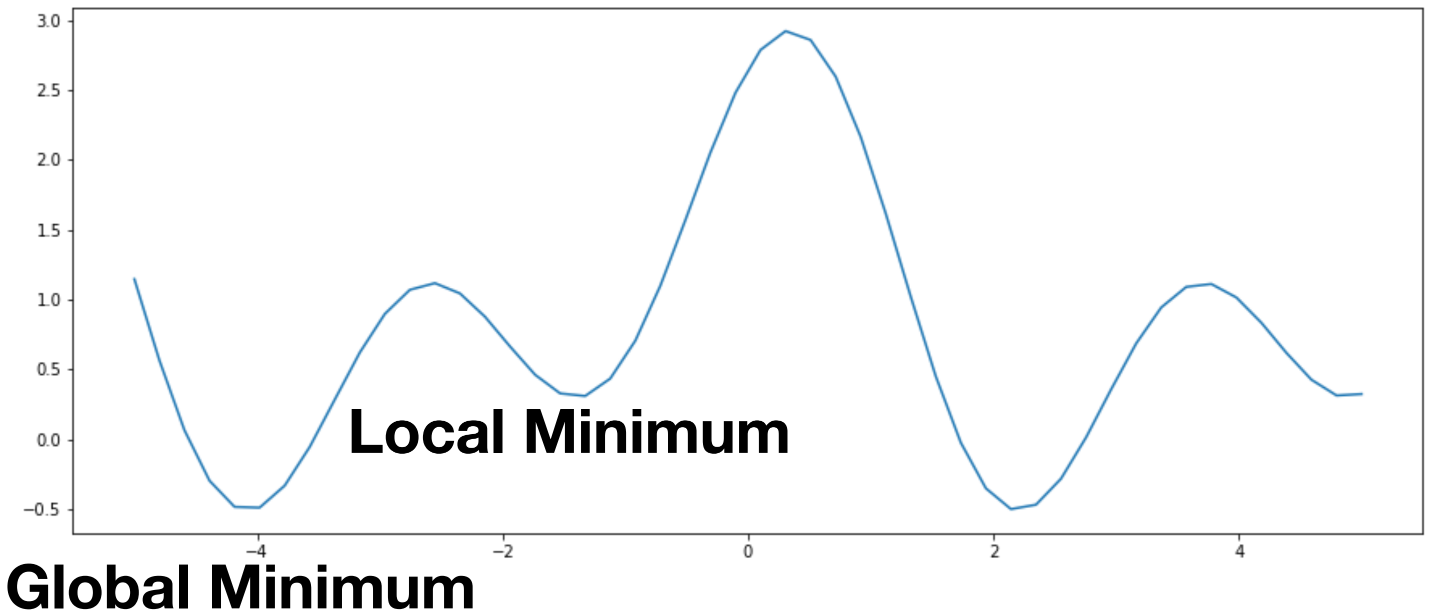
Local Minimum, Local Maximum (2/3)

- Note that the condition on the second derivative is important to distinguish **minimums** from **maximum**



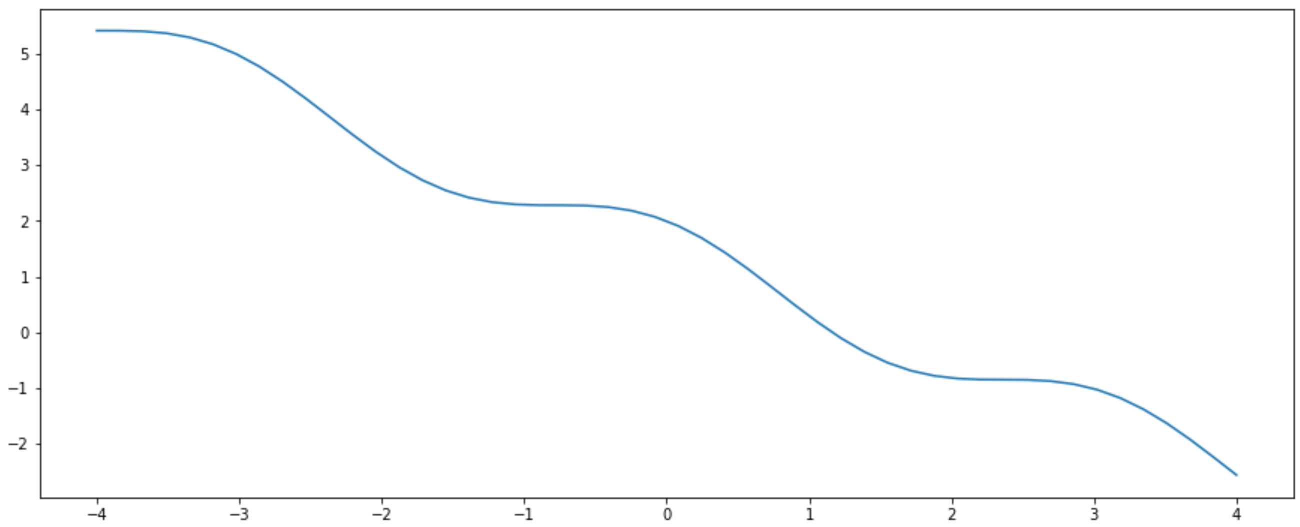
Local Minimum, Local Maximum (3/3)

- Note that the condition on the second derivative is important to distinguish **minimums** from **maximum**. Also, the solution could be only a **local minimum**



Absence of Minimum (1/2)

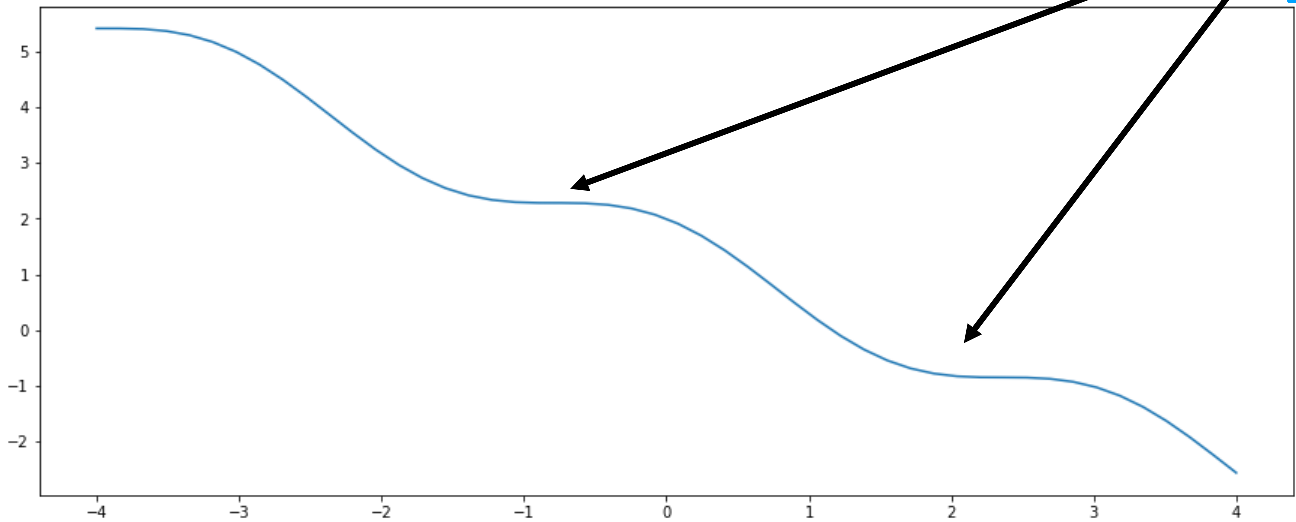
- A function may have no minimum



Absence of Minimum (2/2)

- It is even possible for derivative to be 0 even if the function has no minimum

Derivative is 0 at these points



Let's Take 15 Minutes to Review Derivatives

https://www.maxpixel.net/Fear-Horror-Panic-Comic-Fright-Retro-Face-Man-1296118

Review Derivatives

- Does everybody remember how to compute derivatives?
- Do not panic if you don't.
  - In practice, we will have functions that can compute the derivatives automatically for us
  - Still, you should understand at least how they work
  - We will review briefly the basics

Different Ways of Considering Derivatives

- We can see derivatives in different ways.
- In high school, derivatives are often introduced as a set of rules that let you compute a derivative from a function.
- Let us review that first.

Computing Derivatives

f(x)	f'(x)	
sin(x)	cos(x)	
cos(x)	-sin(x)	
x^n	nx^{n-1}	
ln(x)	$\frac{1}{x}$	
e^x	e^x	
$g(h(x))$	$h'(x)\times g'(h(x))$	**Composition rule**
$g(x)\times h(x)$	$g'(x)\times h(x) + g(x)\times h'(x)$	**Leibniz rule**
$g(x) + h(x)$	$g'(x) + h'(x)$	**Linearity I**
$\alpha \cdot h(x)$	$\alpha \cdot h'(x)$	**Linearity II**

Exercise

Compute the derivatives of the follows and submit it in pdf via PandA by next lecture

- $sin(x) + ln(x)$
- $2 \times ln(x + 1)$
- $sin(2x)$
- $\frac{e^x}{x}$

Different Ways of Considering Derivatives

- We can see derivatives in different ways.
- In high school, derivatives are often introduced as a set of rules that let you compute a derivative from a function.
- Let us review that first.
- The other way to see a derivative is as a <u>local linear approximation of a function</u>

What is a Derivative?

- One definition: the coefficient of the best <u>linear approximation</u> of a function at x

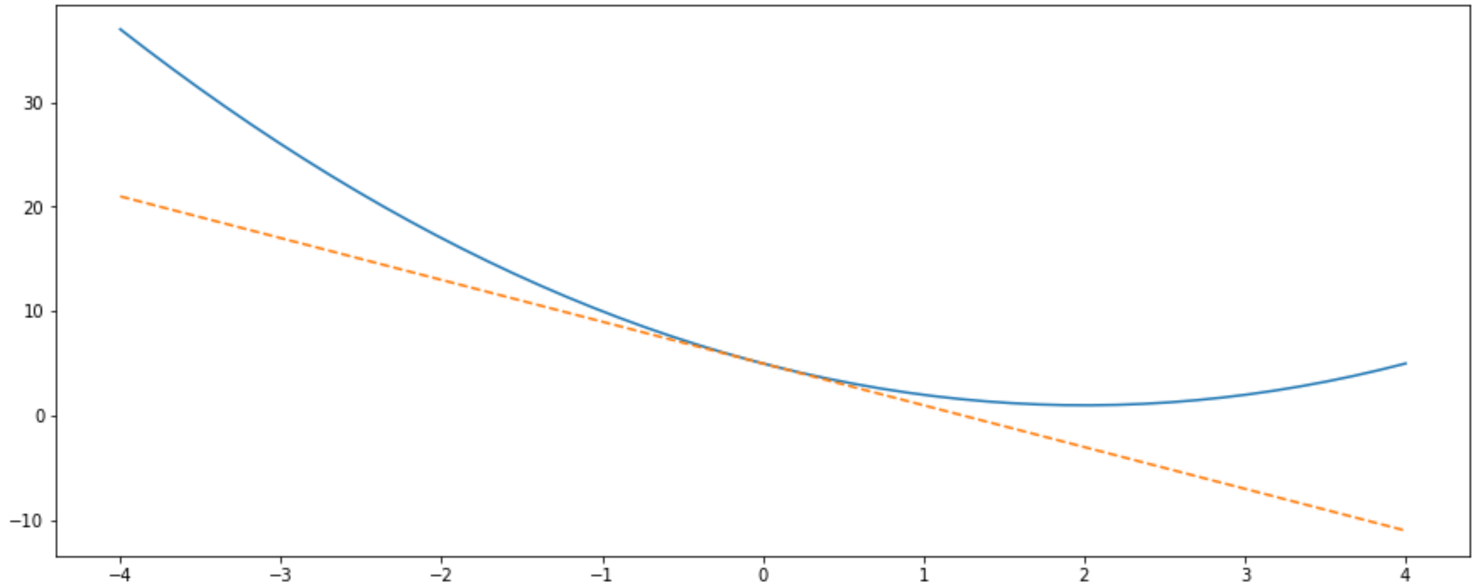
- If h is small: $f(x+h) \approx f(x) + h \cdot f'(x)$
- Example:
  - If we know that $ln(2.3) = 0.832909\ldots$
  - How much is $ln(2.4)$?
    - Supposing we cannot compute a log again
  - 2.4 = 2.3 + 0.1
  - We can approximate: $ln(2.4) \approx ln(2.3) + 0.1 \times \frac{1}{2.3}$
  - Which gives: $ln(2.3) + 0.1 \times \frac{1}{2.3} = 0.876387\ldots$
  - The true value is: $ln(2.4) = 0.875468\ldots$

Different Ways of Considering Derivatives

- We can see derivatives in different ways.
- In high school, derivatives are often introduced as a <u>set of rules</u> that let you compute a derivative from a function.
- Let us review that first.
- The other way to see a derivative is as a <u>local linear approximation of a function</u>
- Equivalently, the derivative is the <u>slope of the tangent</u> of the function at a point

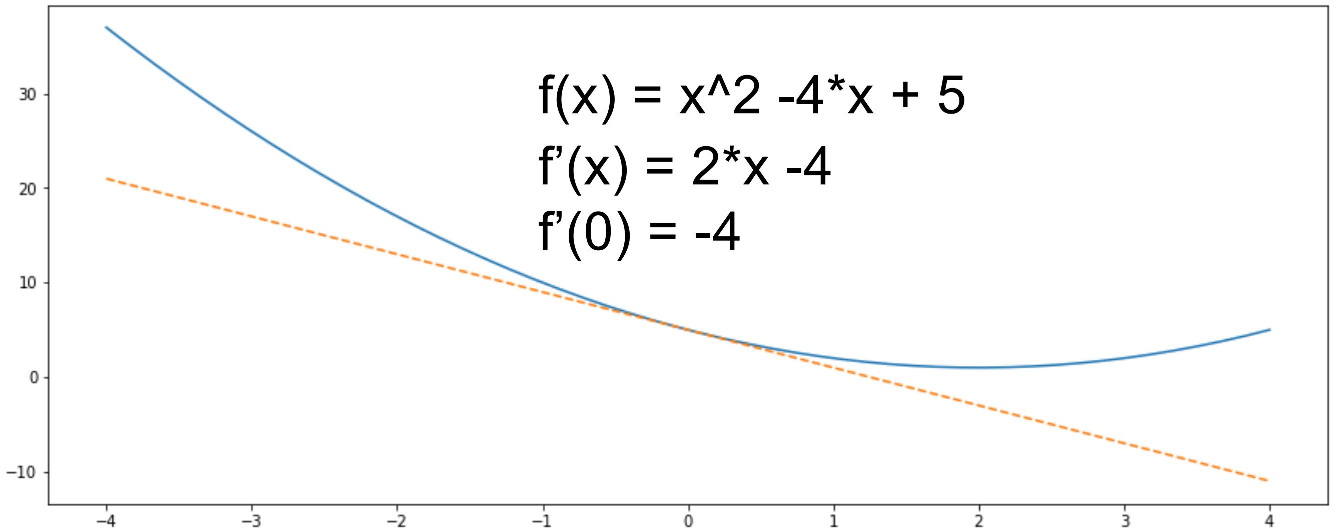
What is a Tangent?

- The line that best approximates a line at a point



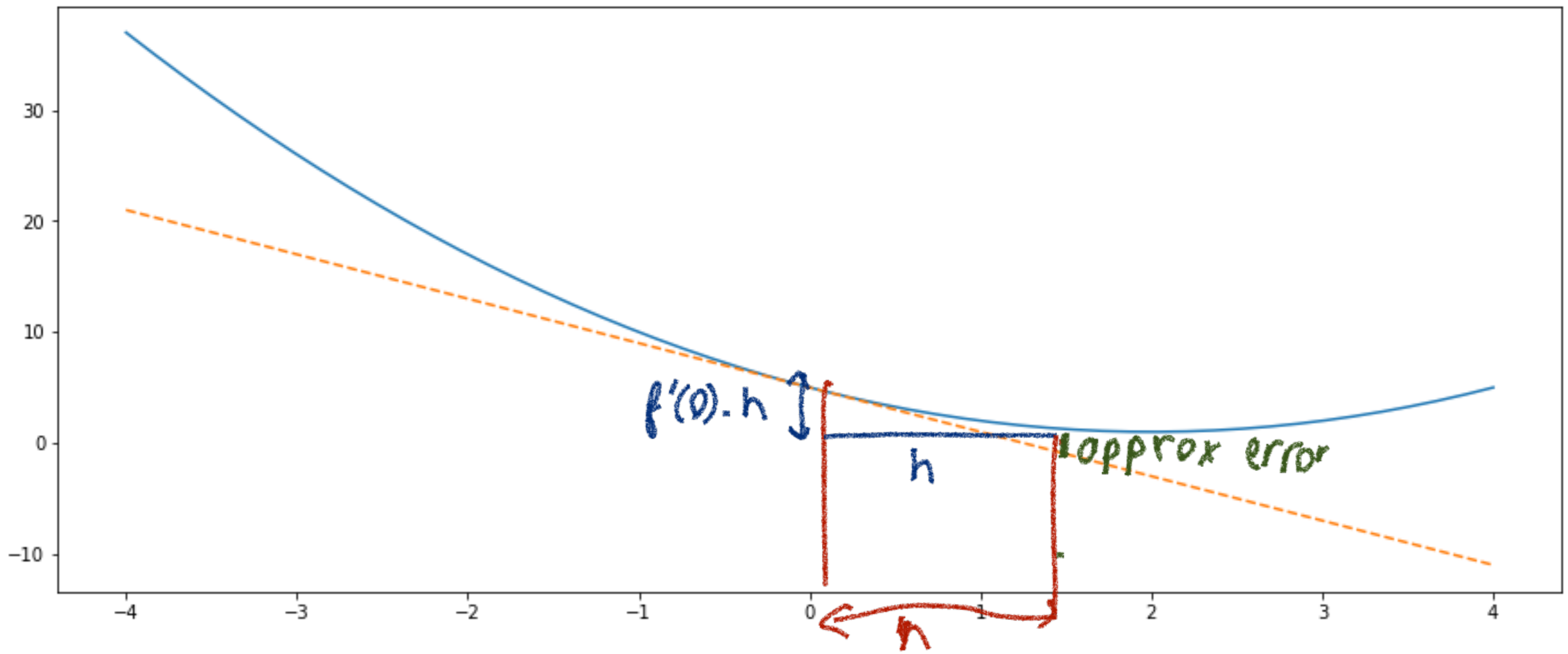
What is a Derivative?

- The derivative is also the coefficient of the tangent to the graph of the function.



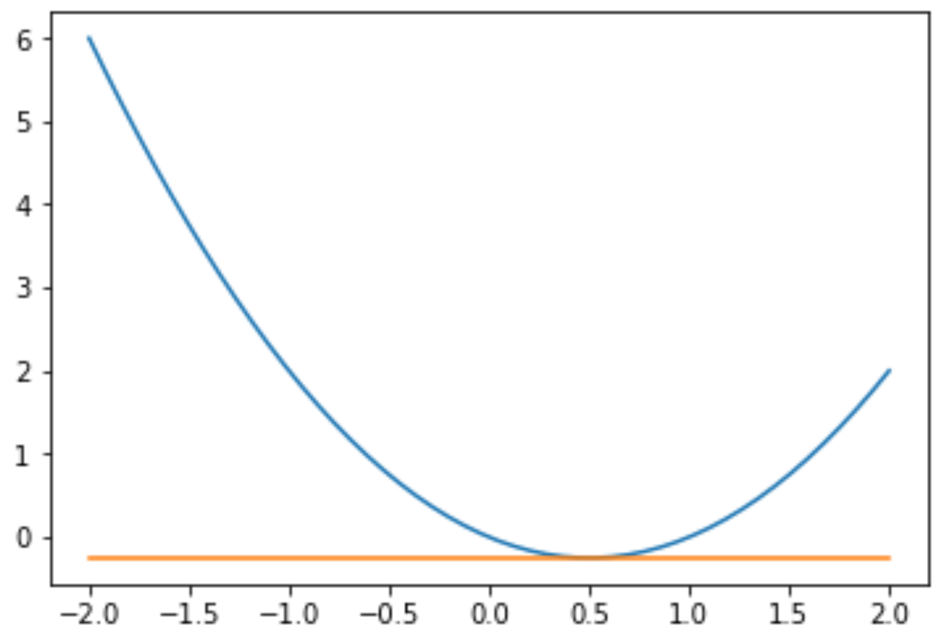
Tangent and Derivative

$$f(x+h) \approx f(x) + h \cdot f'(x)$$



Derivative and Minimum (1/3)

- Intuitively, this shows you why the derivative should be zero at a minimum

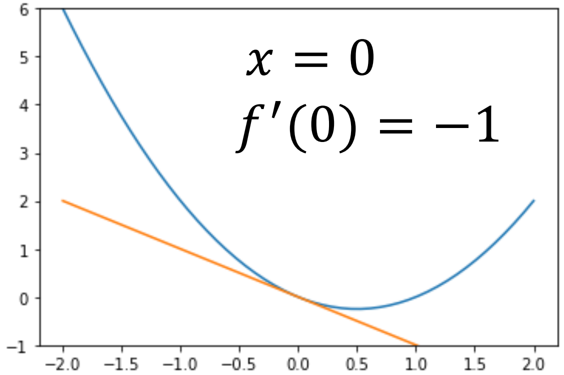


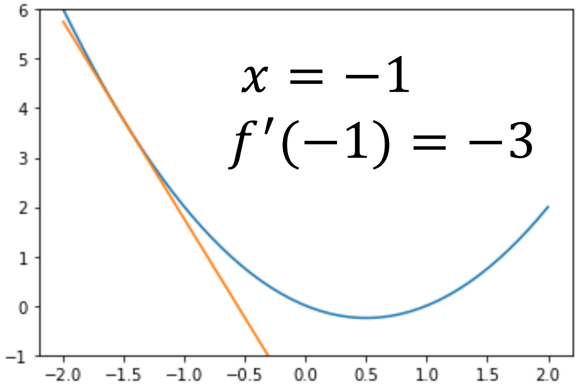
Derivative and Minimum (2/3)

- Let us look again at how a derivative can help us find a minimum

Derivative and Minimum (3/3)

- The derivative tells us in which direction move to find the minimum

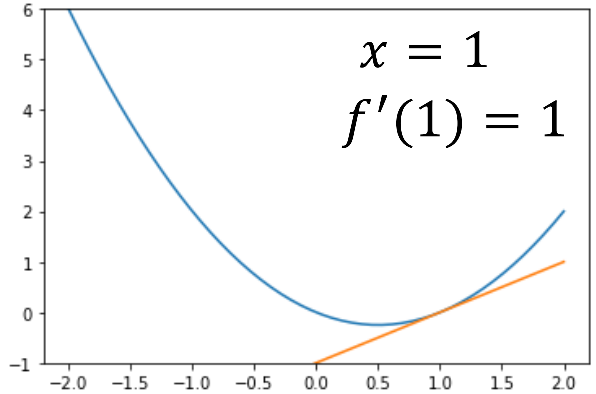


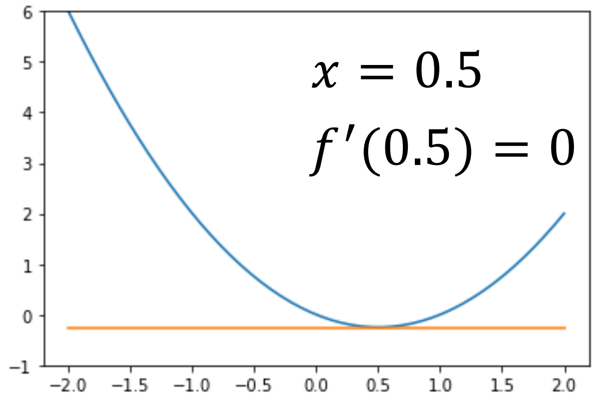


If derivative at x is negative, minimum is on the right

If derivative at x is positive, minimum is on the left

If derivative at x is zero, x should be a minimum





Gradient Descent Algorithm (1/5)

- This suggests some procedure for finding a minimum:
  - Start at any x (e.g., x = 0)
  - Compute f'(x)
    - If f'(x) > 0: Decrease x a bit
    - If f'(x) < 0: Increase x a bit
  - Repeat

Gradient Descent Algorithm (2/5)

- This suggests some procedure for finding a minimum:
  - Start at any x (e.g., x = 0)
  - Compute f'(x)
    - If f'(x) > 0: Decrease x a bit
    - If f'(x) < 0: Increase x a bit
  - Repeat

In practice, we do this:

$$x := x - lr \cdot f'(x)$$

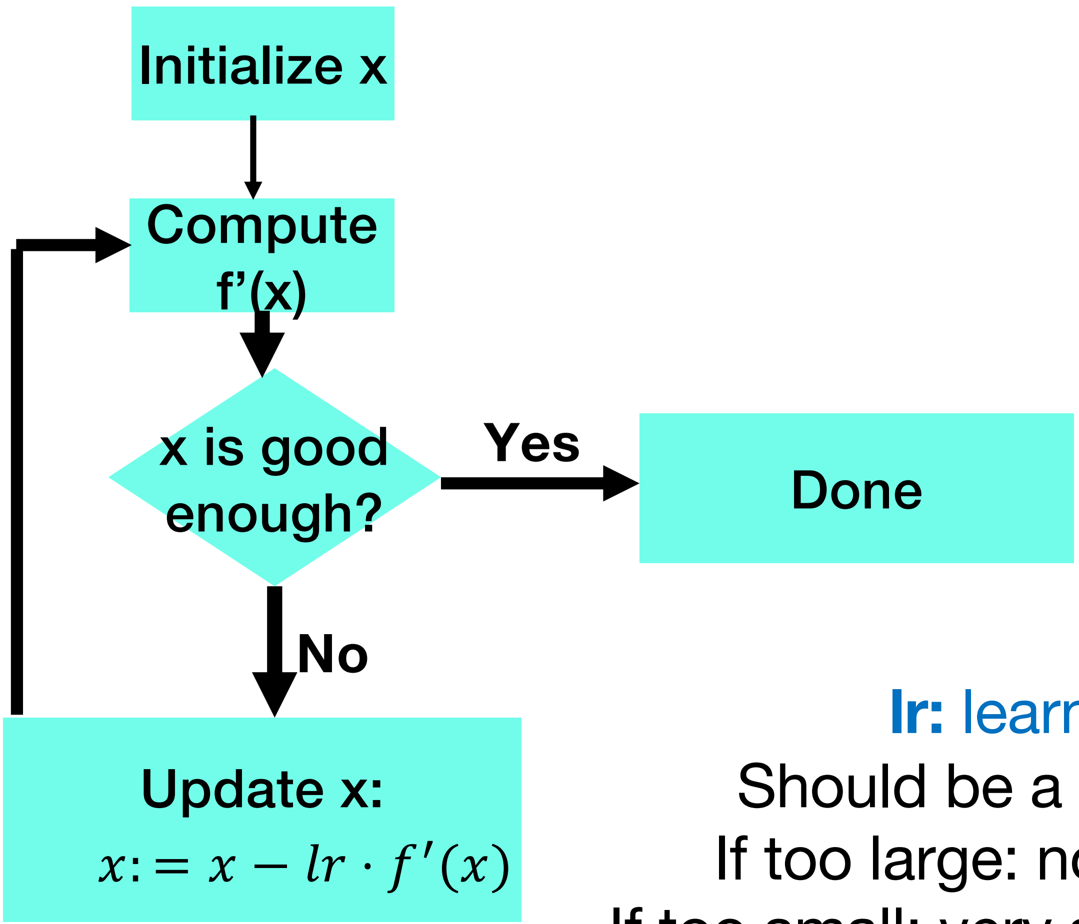
lr: learning rate

Should be a positive value

If too large: no convergence

If too small: very slow convergence

Gradient Descent Algorithm (3/5)



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

$$f'(x) = 2x - 1$$

$$\underset{x}{\operatorname{argmin}} f(x) = 0.5$$

lr = 0.2

x = 0
f'(x) = -1

x=0.2
f'(x) = -0.6

x=0.32
f'(x) = -0.36

x = 0.392
…
…
x=0.493
f'(x) = -0.014

STOP?

lr: learning rate
Should be a positive value
If too large: no convergence
If too small: very slow convergence

Gradient Descent Algorithm (4/5)

- Gradient descent works well **even** when we have functions of millions of variable
  - This is why it is so useful for Machine Learning and Neural Networks
  - Other methods will not be practical in such settings
- Convergence will depend on the choice of a good **learning rate**
  - In experiments, a good deal of time is often spent finding an optimal learning rate
  - **Too large** learning rate: **no** convergence (i.e., the system learn nothing)
  - **Too small** learning rate: **slow** convergence (i.e., the system takes a long time to learn)

Gradient Descent Algorithm (5/5)

- Let us try to see a bit more how it works in practice using Jupiter Notebooks

https://shorturl.at/NIfVv