Mathematical Concepts

Functions, Minimization, Gradient

Fundamentals of Artificial Intelligence
Fabien Cromieres
Kyoto University

bit.ly/2v6OjgT

What we are going to study/review

- Functions of one variable
- Functions of several variables
- Derivatives and Gradient
- Finding the minimum of a function with Gradient Descent

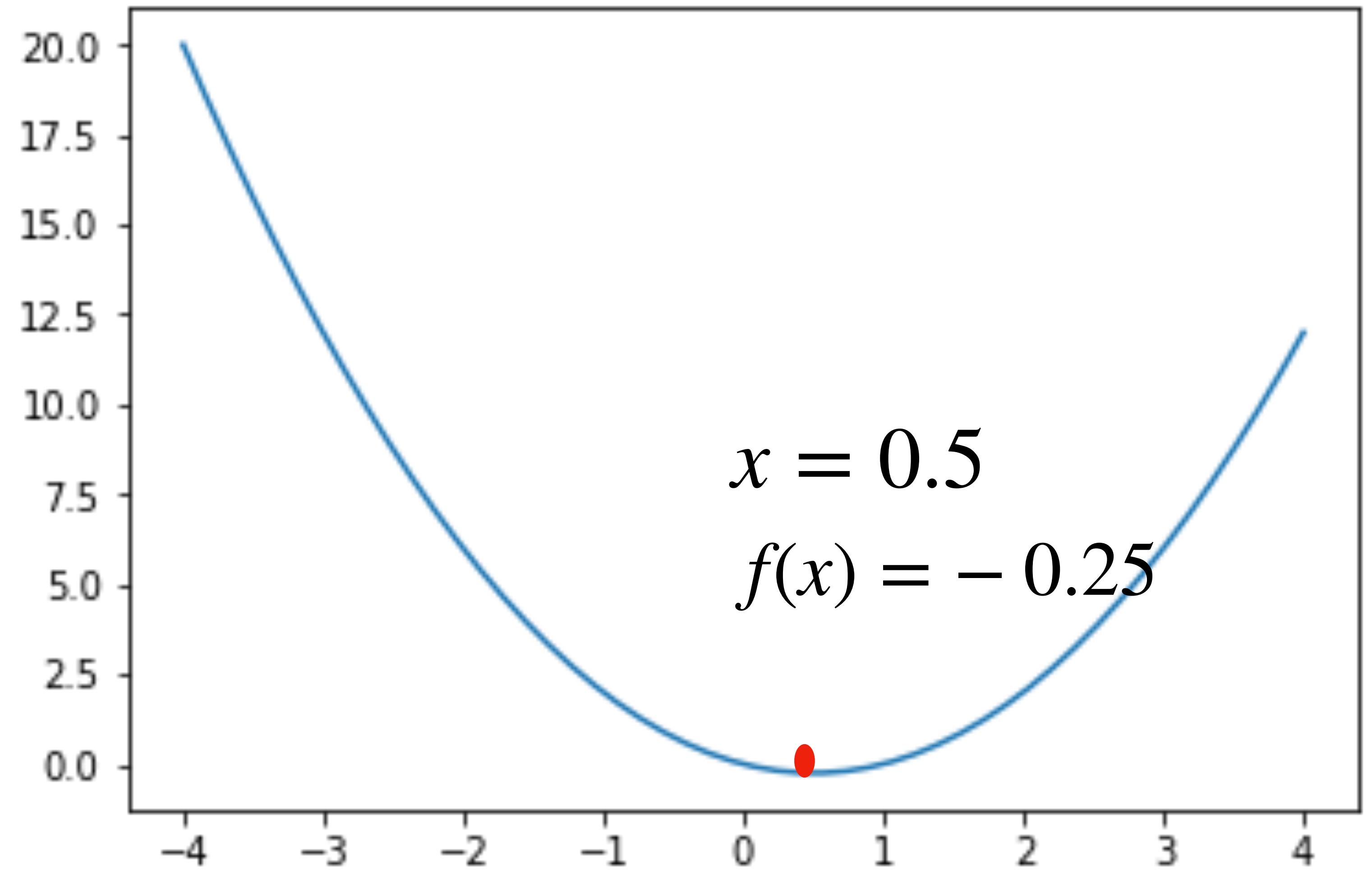
What we are going to study/review

- 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

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

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

$$\underset{x}{\arg\min} f(x)$$



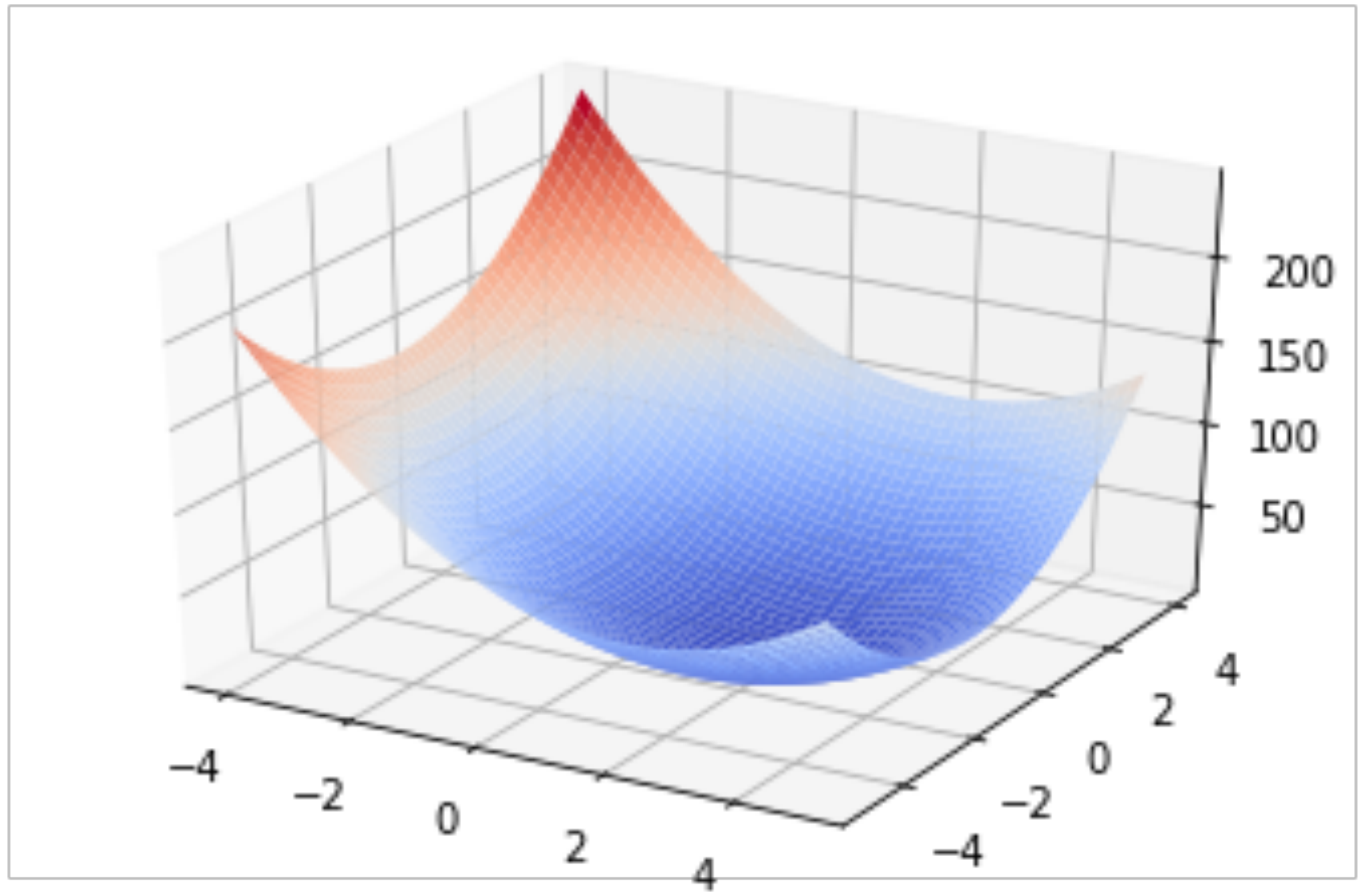
What we are going to study/review

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

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

$$\arg\min_{x,y} f(x, y)$$

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

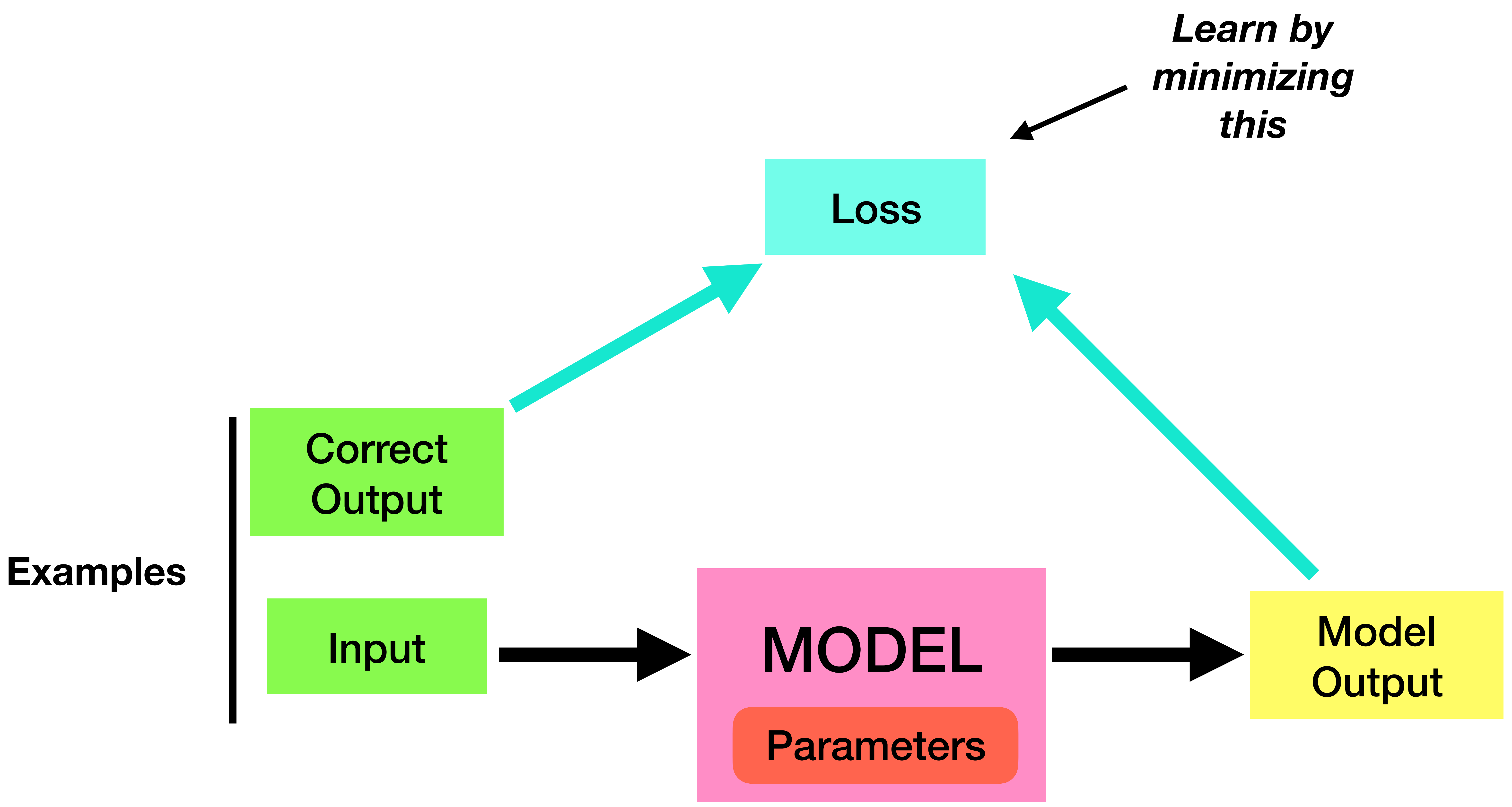


Why we do it?

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

Supervised Learning

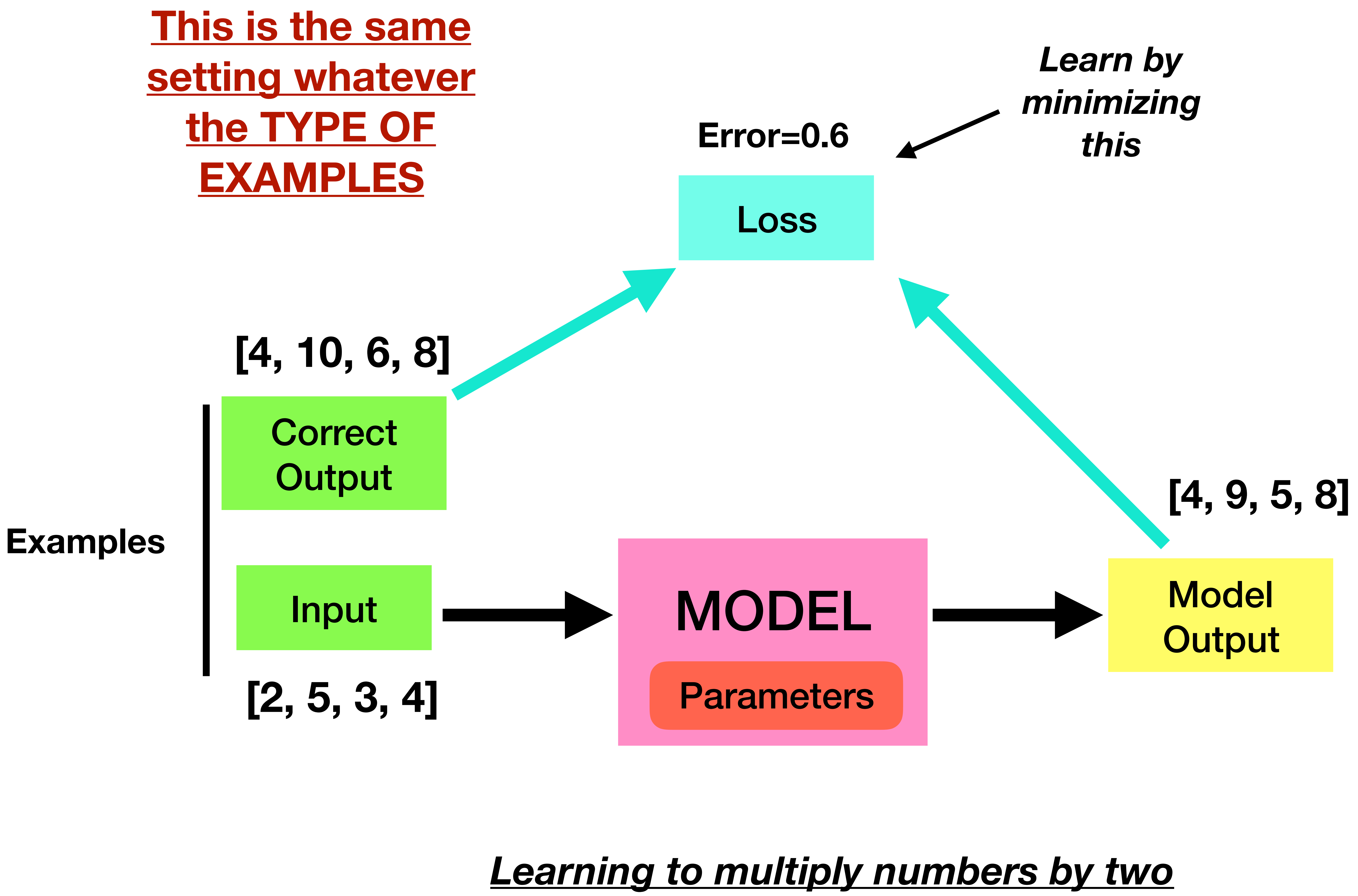
- In supervised learning, we usually have:
  - A **MODEL**: a "parameterized" function that takes input and produce output
  - A ***Loss***: A function that compute how different the model output is from the correct output
  - ***Examples*** of input and correct output



A Peek in the Future

Supervised Learning

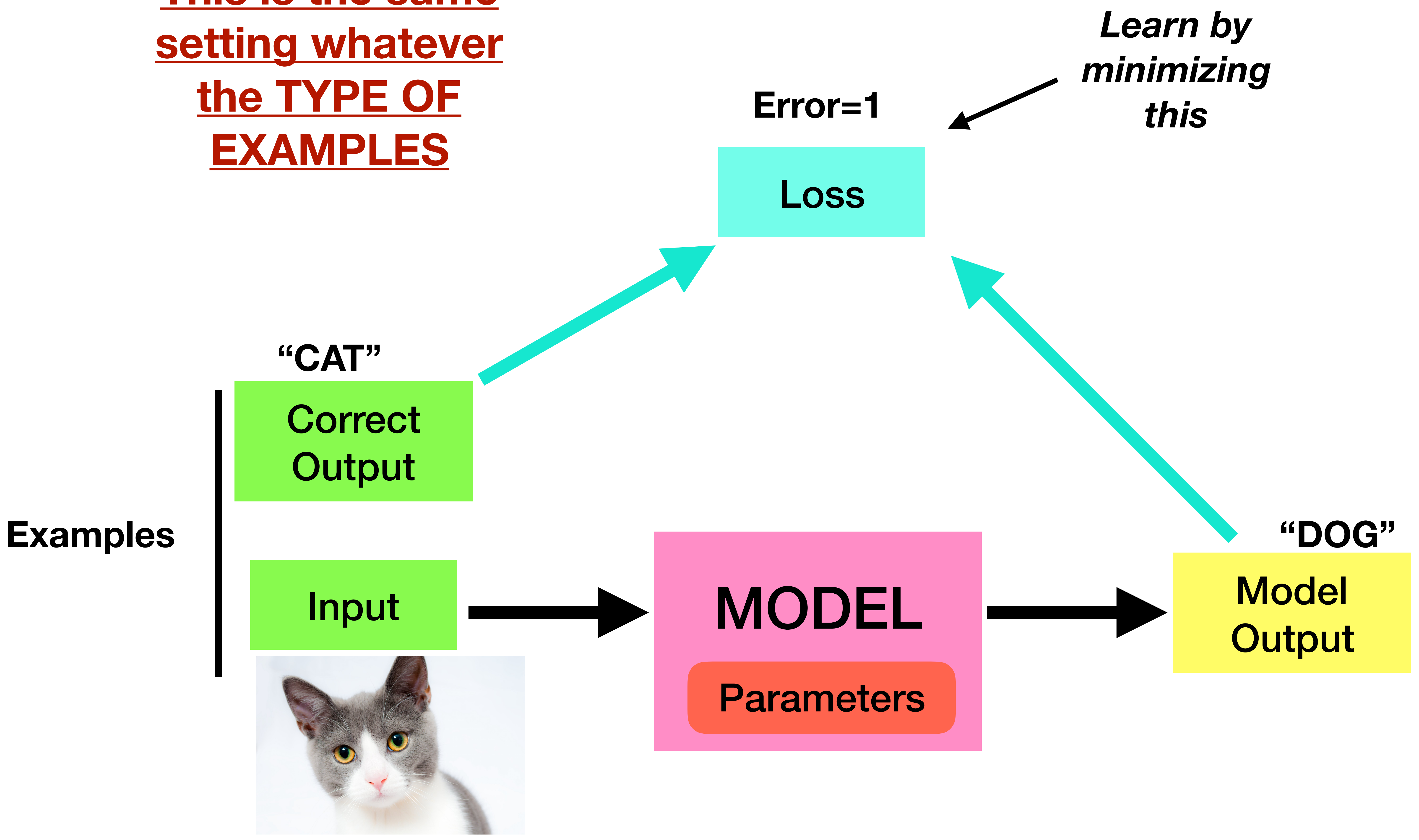
- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that compute how different the model output is from the correct output
  - ***Examples*** of input and correct output



Supervised Learning

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

<u>This is the same setting whatever the TYPE OF EXAMPLES</u>

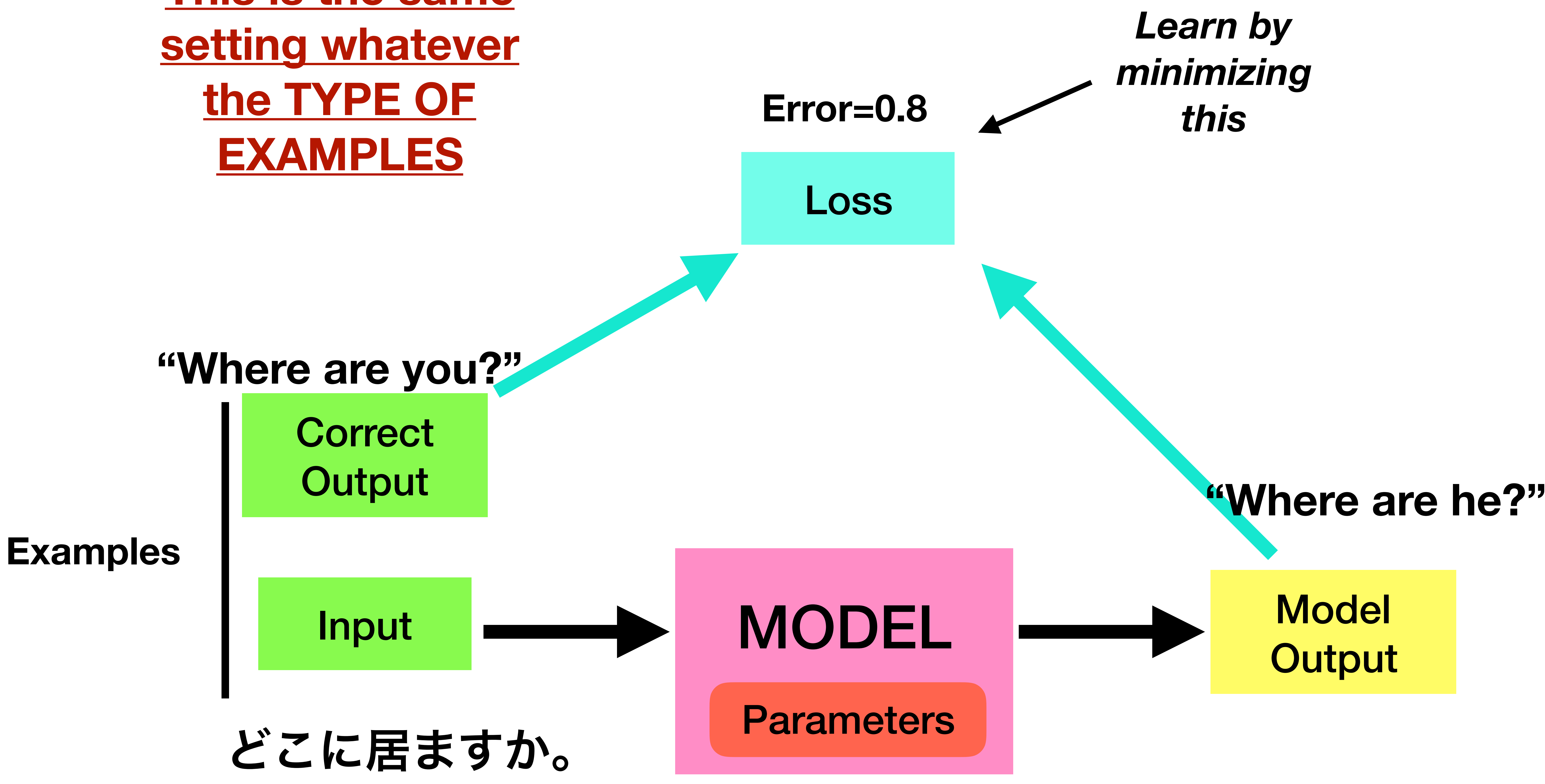


<u>Learning to recognize images</u>

Supervised Learning

- In supervised learning, we usually have:
  - A **MODEL**: a “parameterized” function that takes input and produce output
  - A ***Loss***: A function that compute 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

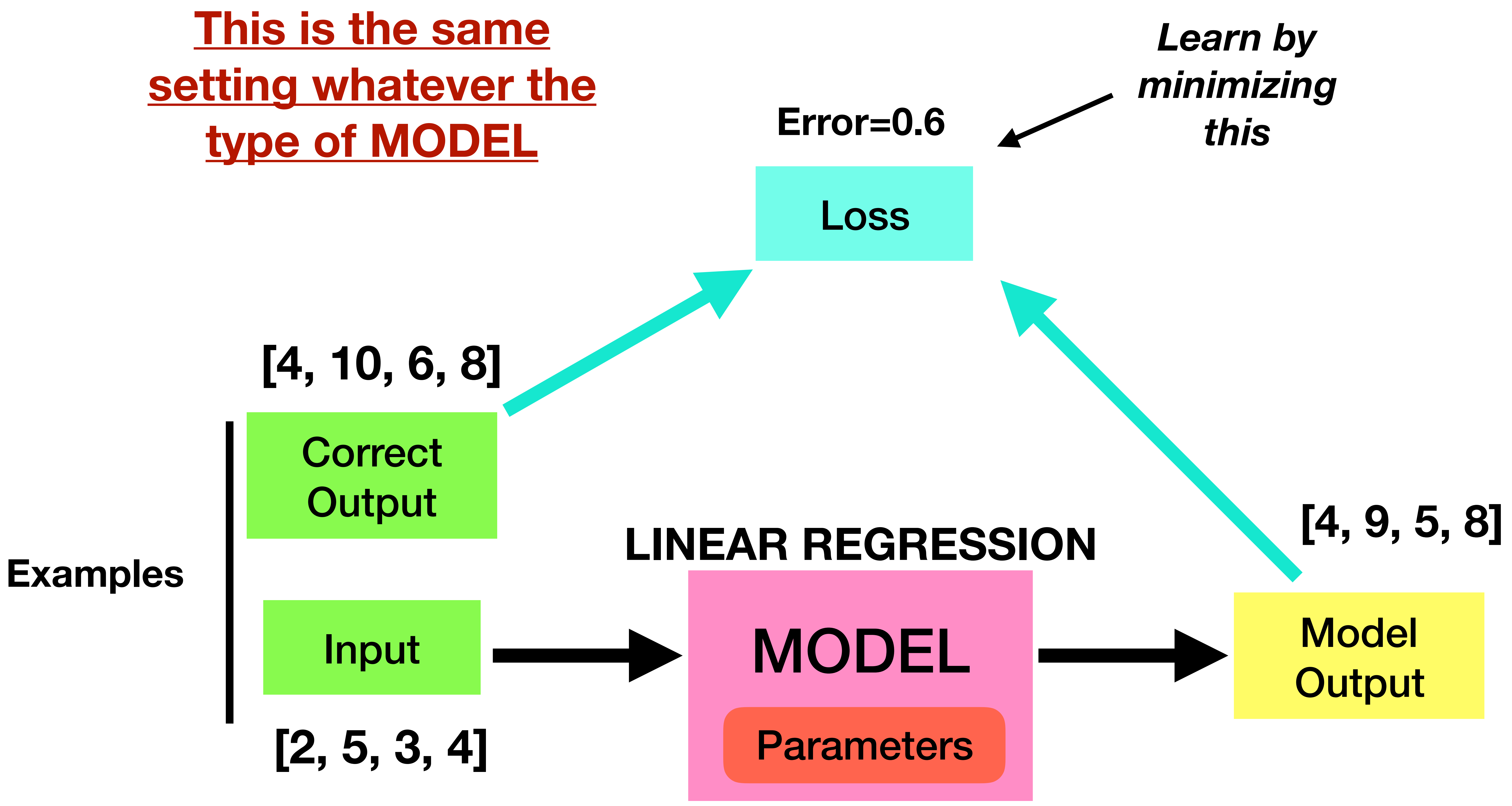


Learning to translate

A Peek in the Future

Supervised Learning

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

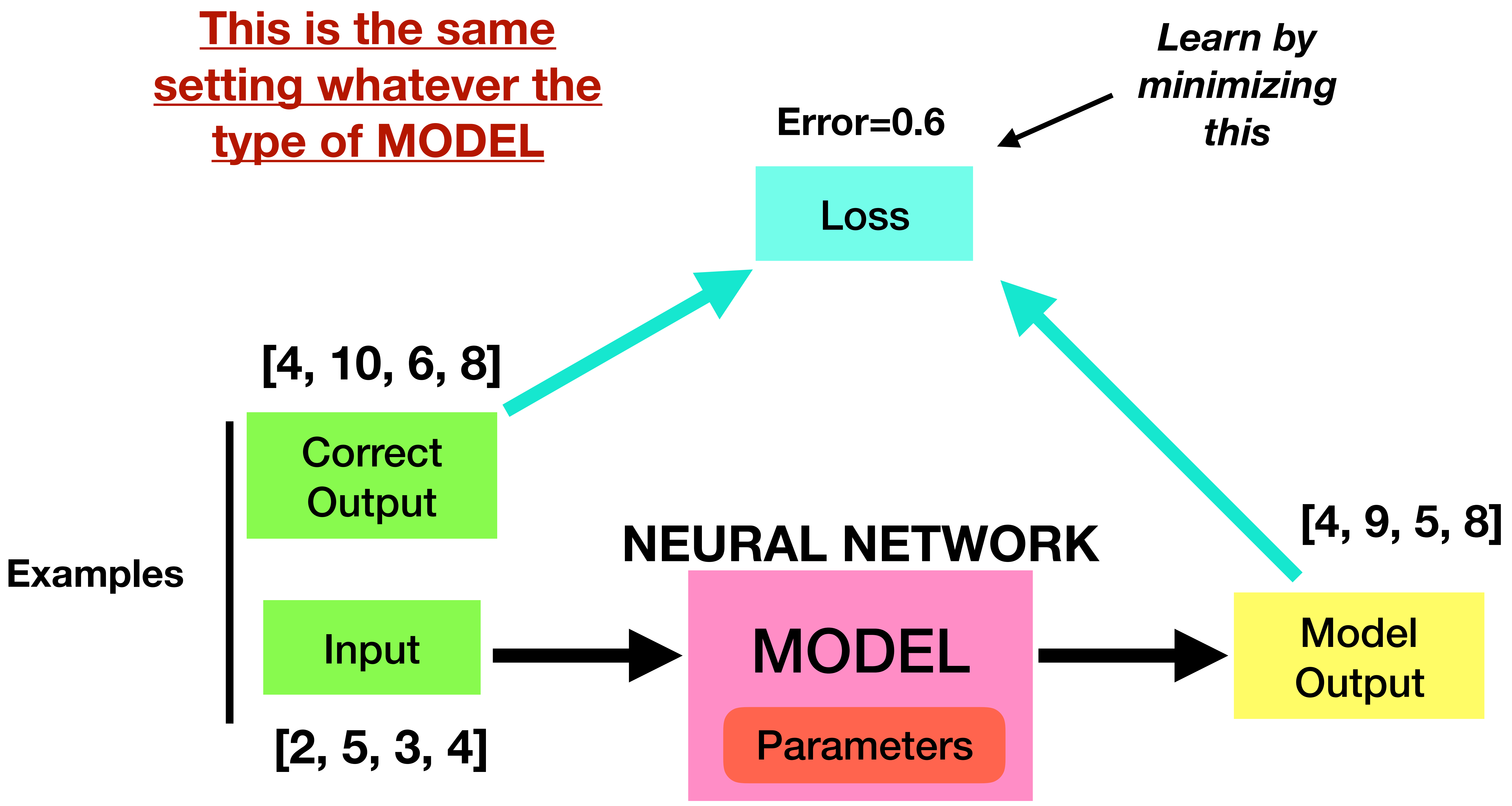


Learning to multiply numbers by two with a Linear Regression Model

A Peek in the Future

Supervised Learning

- In supervised learning, we usually have:
  - A **MODEL**: a "parameterized" function that takes input and produce output
  - A ***Loss***: A function that compute 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

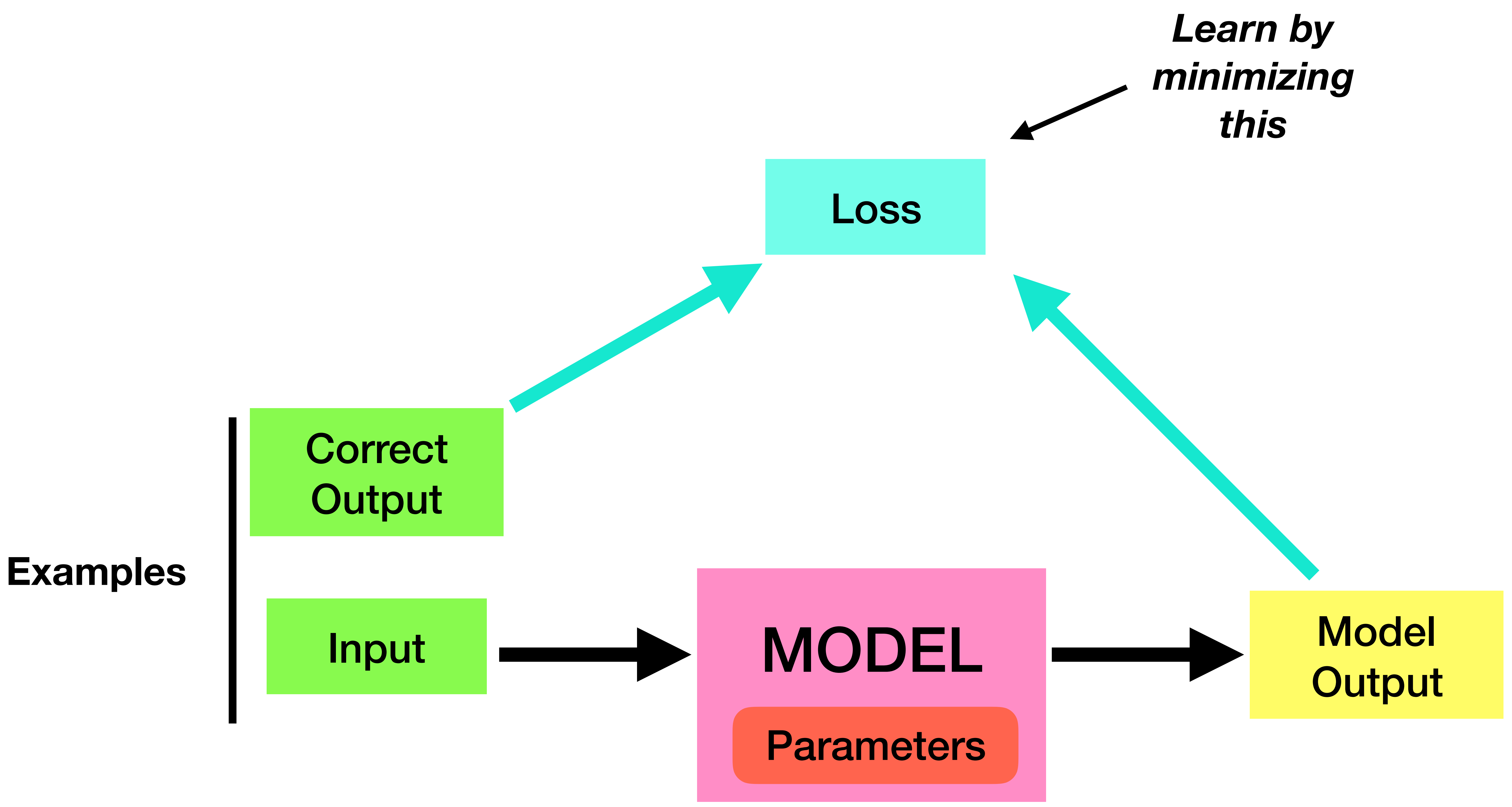
Terminology

- 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

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

$$f(-1) = 2$$
$$f(0) = 0$$
$$f(0.1) = -0.99$$

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 $\arg\min_x f(x)$
- What is $\arg\min_x f(x)$ for $f(x) = x^2 + 3$?
- What is $\arg\min_x f(x)$ for $f(x) = x$?
- What is $\arg\min_x f(x)$ for $f(x) = x^2 - x$?

Minimizing a function of one variable

pollev.com/fabiencromie576

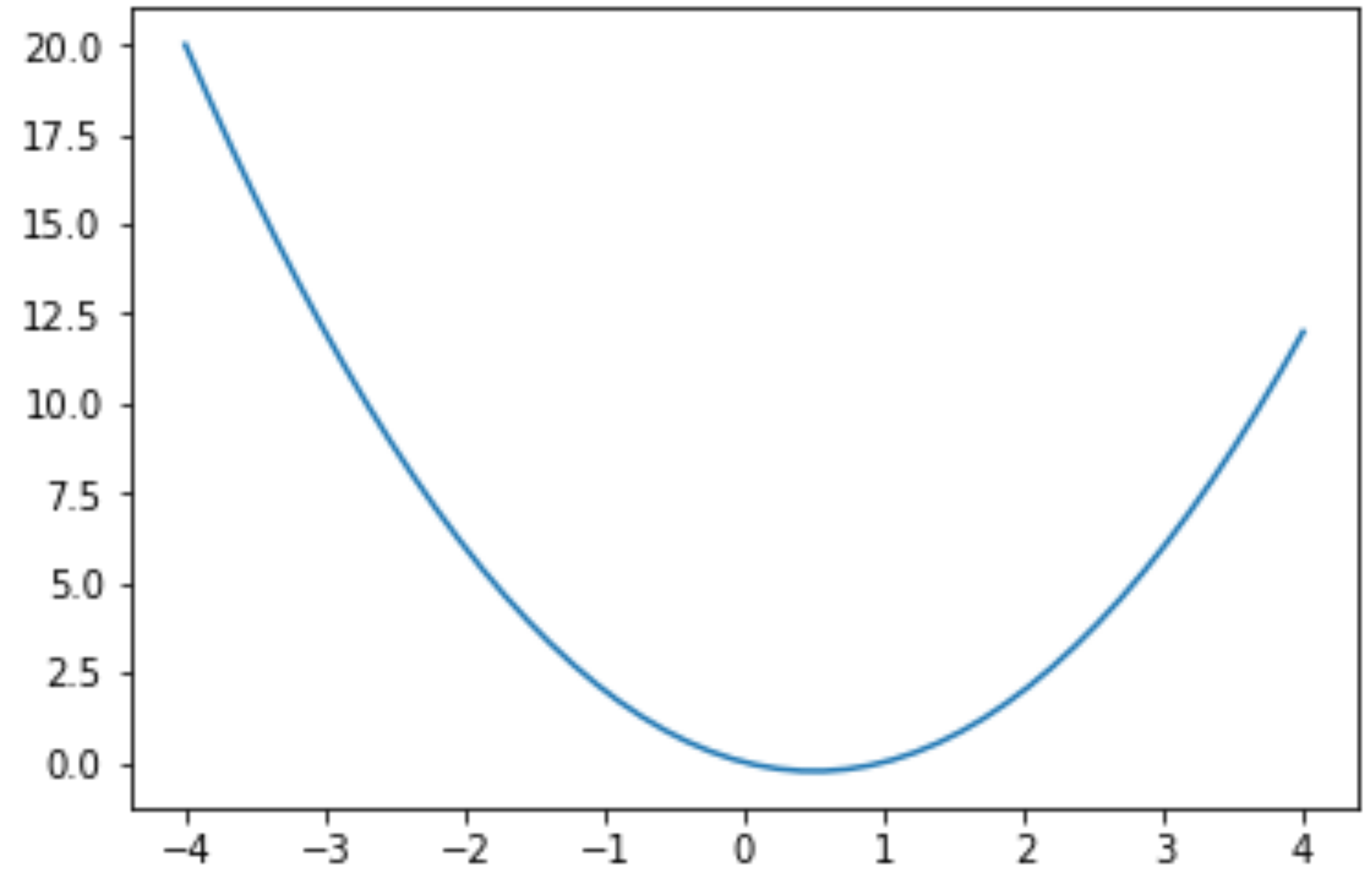
- Given a function of one variable ***f(x)***, what is the input number ***x*** that gives the smallest output number?
- We note this number $\arg\min_x f(x)$
- What is $\arg\min_x f(x)$ for $f(x) = x^2 + 3$?
- What is $\arg\min_x f(x)$ for $f(x) = x$?
- What is $\arg\min_x f(x)$ for $f(x) = x^2 - x$?

The “High School” view of minimization

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

The “High School” view of minimization

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



To minimize f(x):

1. Compute **first derivative** f’(x)

3. Compute **second derivative** f”(x)

5. Find x0 such that f’(x0) = 0

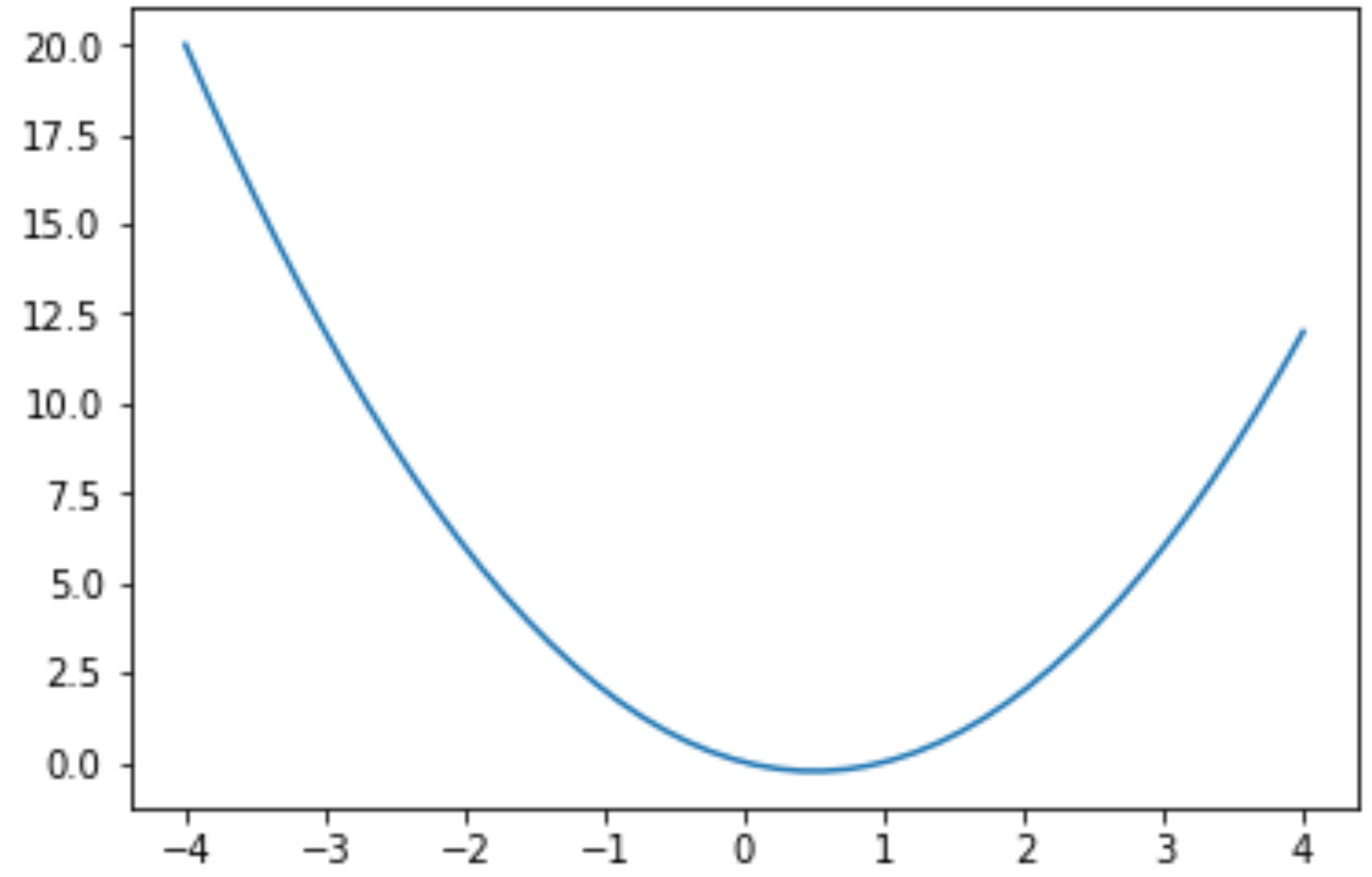
6. If f”(x0) > 0 then x0 is a **local minimum** of f

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

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

The “High School” view of minimization

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



To minimize f(x):

1. Compute **first derivative** f’(x)

3. Compute **second derivative** f”(x)

5. Find x0 such that f’(x0) = 0

6. If f”(x0) > 0 then x0 is a **local minimum** of f

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

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

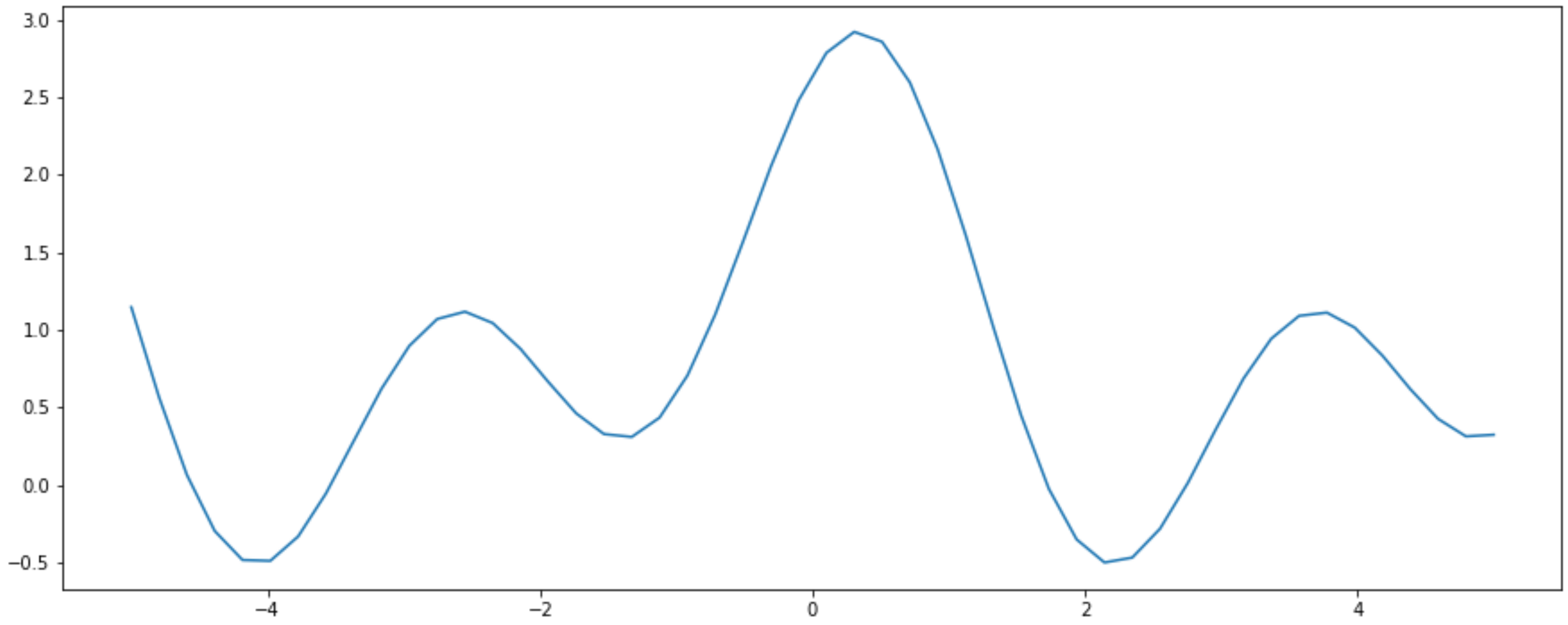
$$f'(x) = 2x - 1 \rightarrow x_0 = 0.5$$

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

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

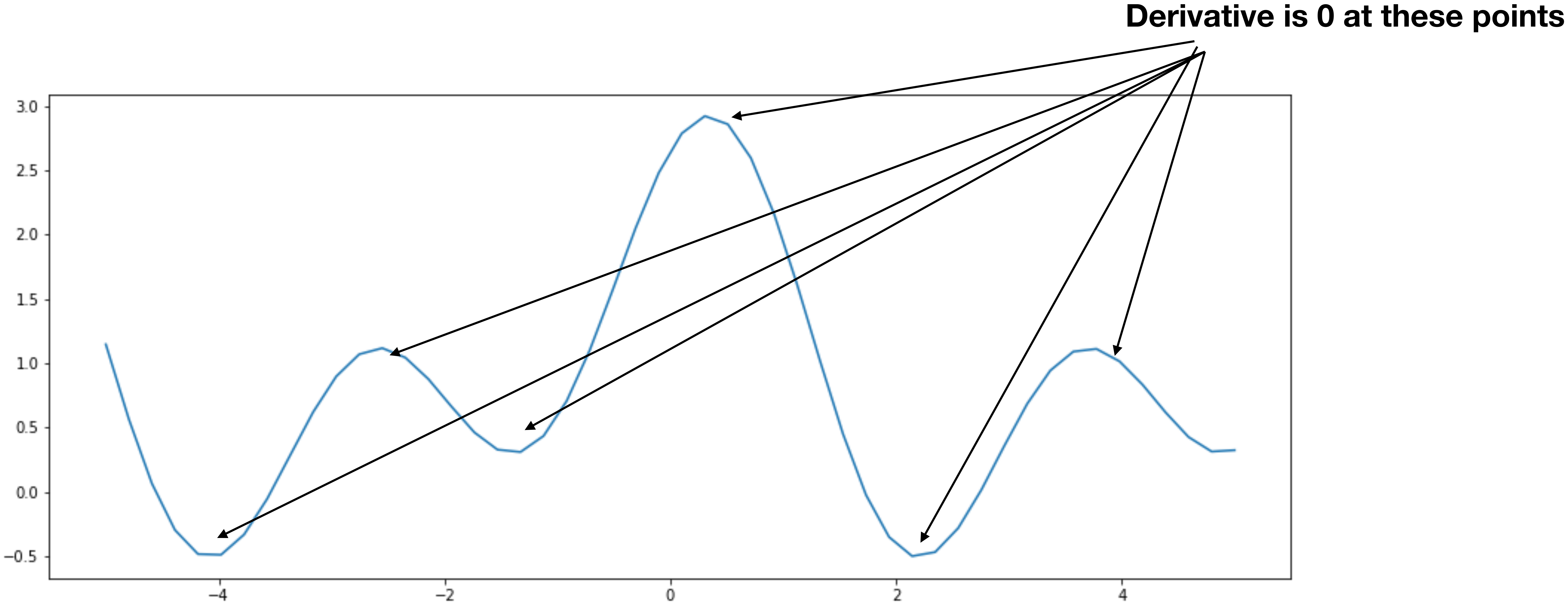
Local minimum, local maximum

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



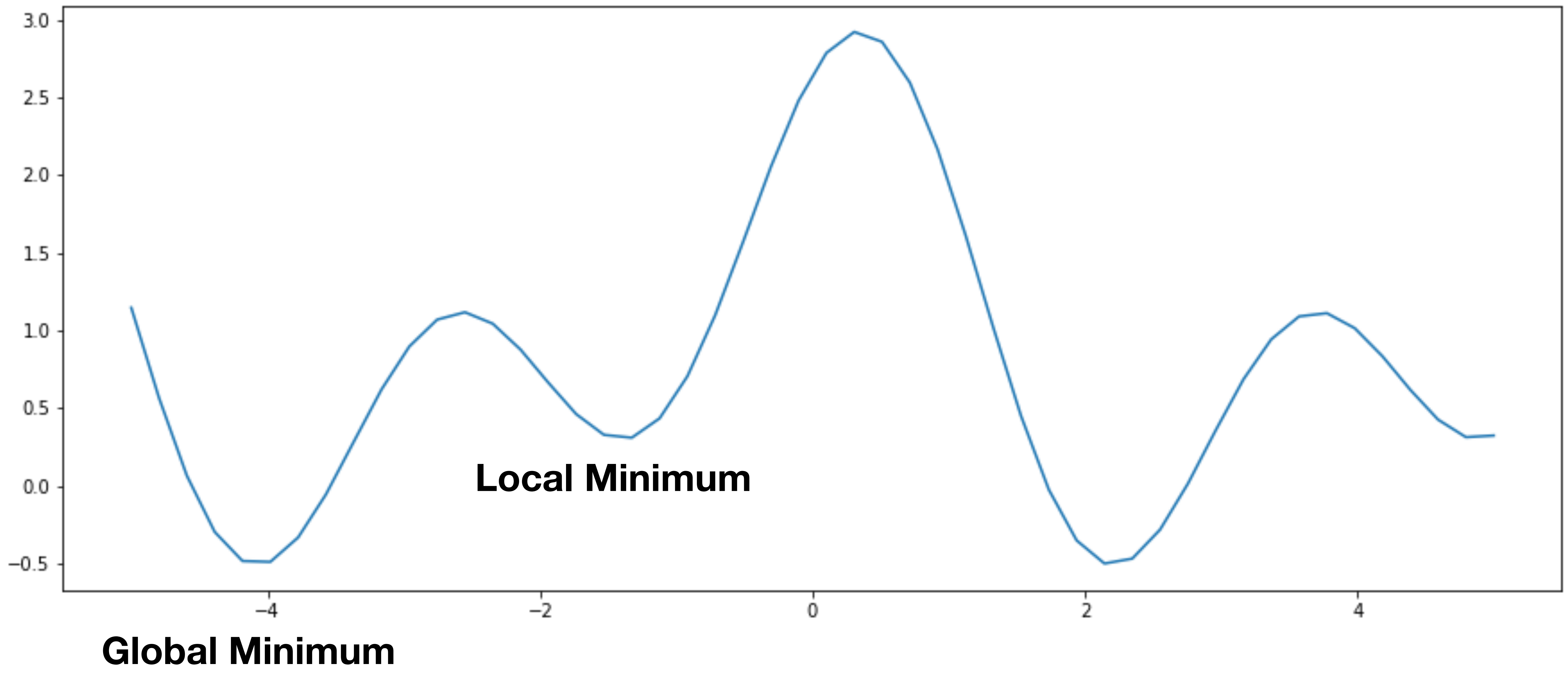
Local minimum, local maximum

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



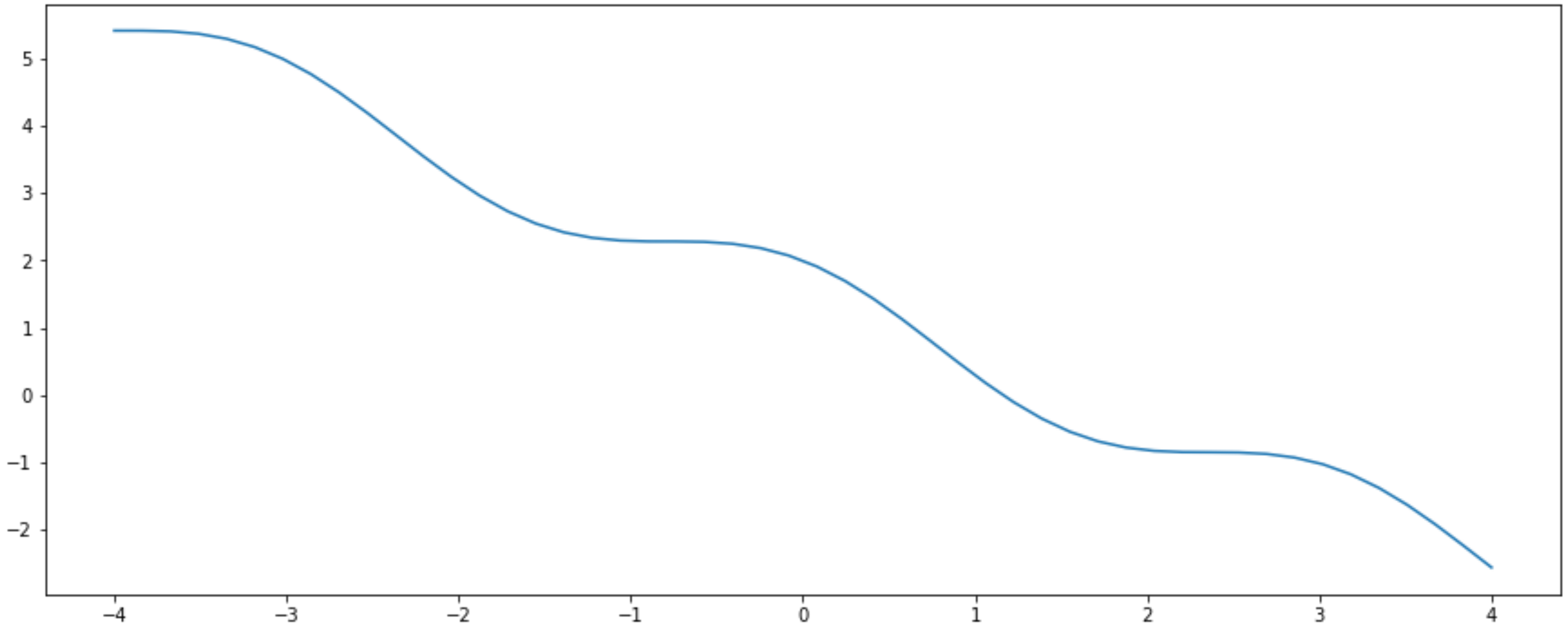
Local minimum, local maximum

- 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

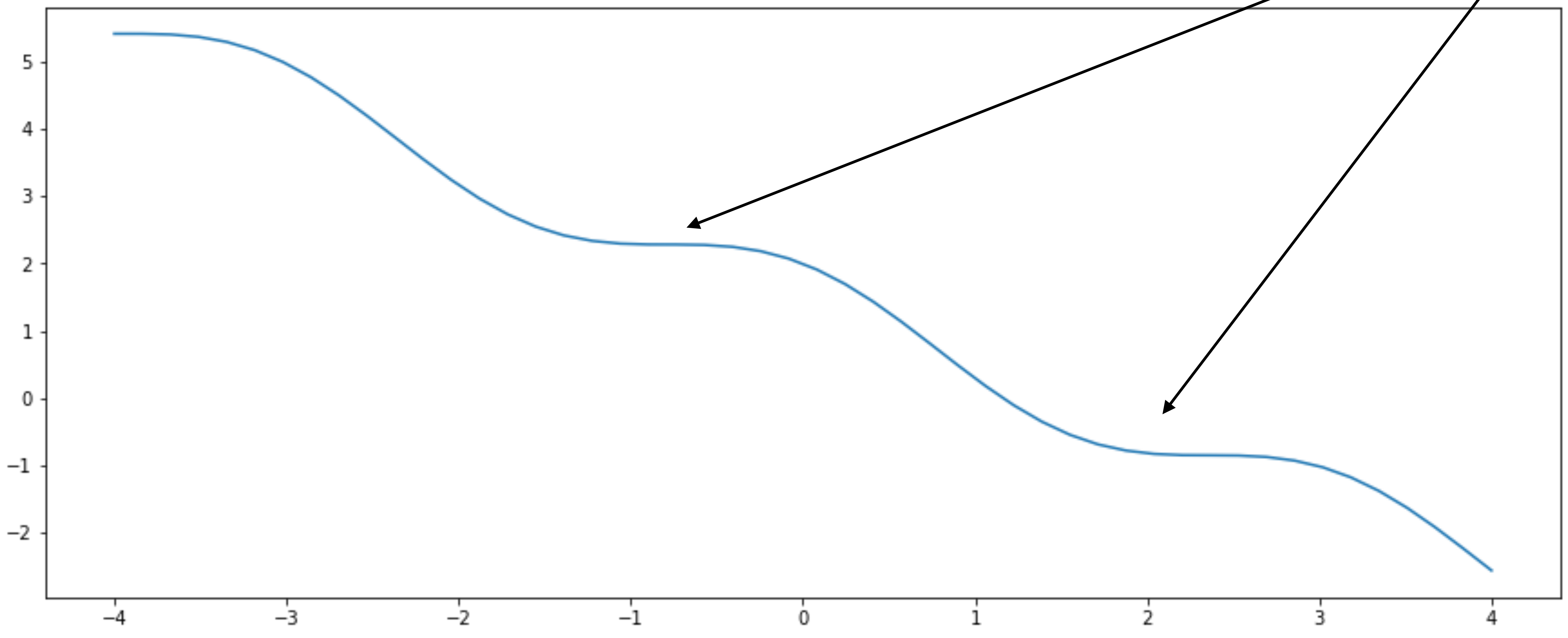
- A function may have no minimum



Absence of minimum

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

Derivative is 0 at these points



Let us take 15 minutes to review derivatives

Derivatives! Oh no!!

Don’t worry. It is a quick review. But in the real world, we rarely have to compute derivatives

Computers do that for us.

Still. It is good to have some basics.

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

Let us take 15 minutes to review derivatives

- Does everybody remember how to compute derivatives?
- <u>Do not panic</u> 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}	
log(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:

$sin(x) + log(x)$

$2 \times log(x+1)$

$sin(2x)$

$\frac{e^x}{x}$

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 <u>definition</u>: the coefficient of the best linear approximation of a function at x
- If h is small: $f(x+h) \approx f(x) + h \cdot f'(x)$
- Example:
  - if we know that $log(2.3) = 0.832909...$
  - How much is $log(2.4)$?
    - Supposing we cannot compute a log again
  - 2.4 = 2.3 + 0.1
  - We can approximate: $log(2.4) \approx log(2.3) + 0.1 \times \frac{1}{2.3}$
  - Which gives: $log(2.3) + 0.1 \times \frac{1}{2.3} = 0.876387...$
  - The true value is: $log(2.4) = 0.875468...$

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.

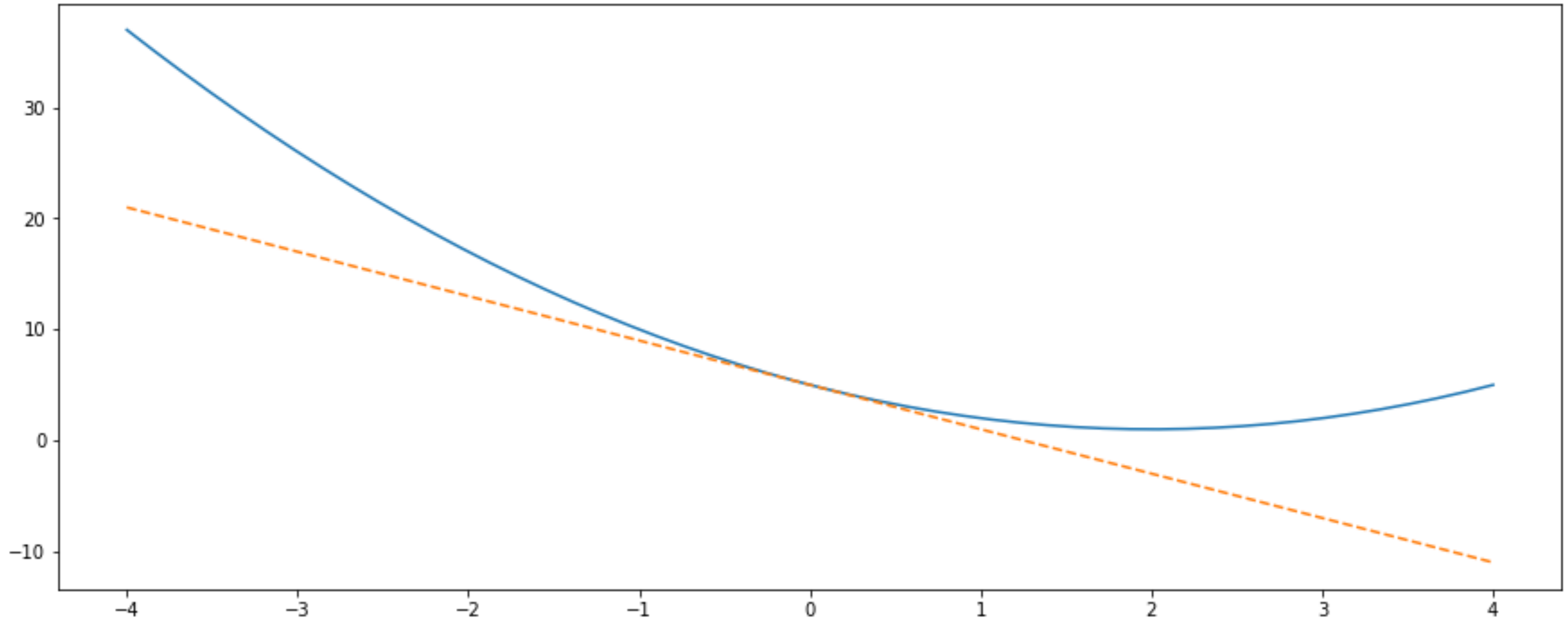
- Let us review that first.

- 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

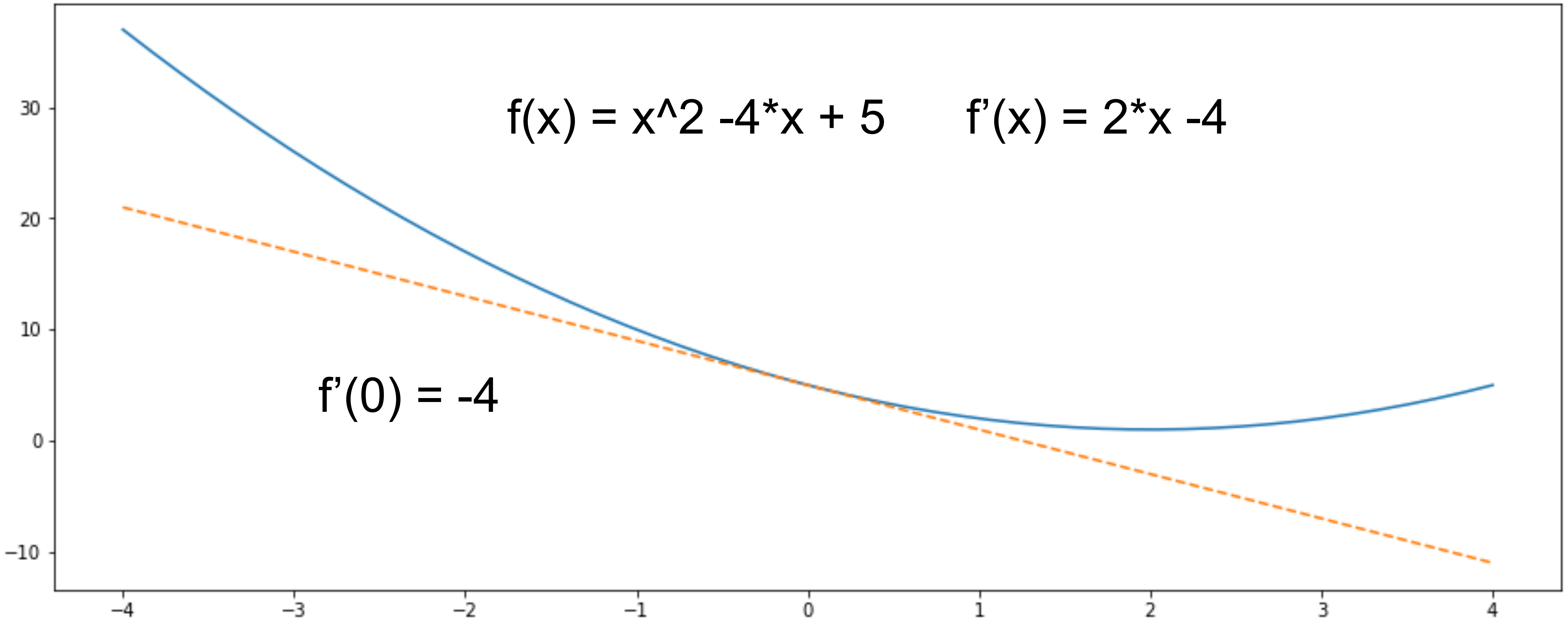
What is a tangent?

- The line that best approximate a line at a point



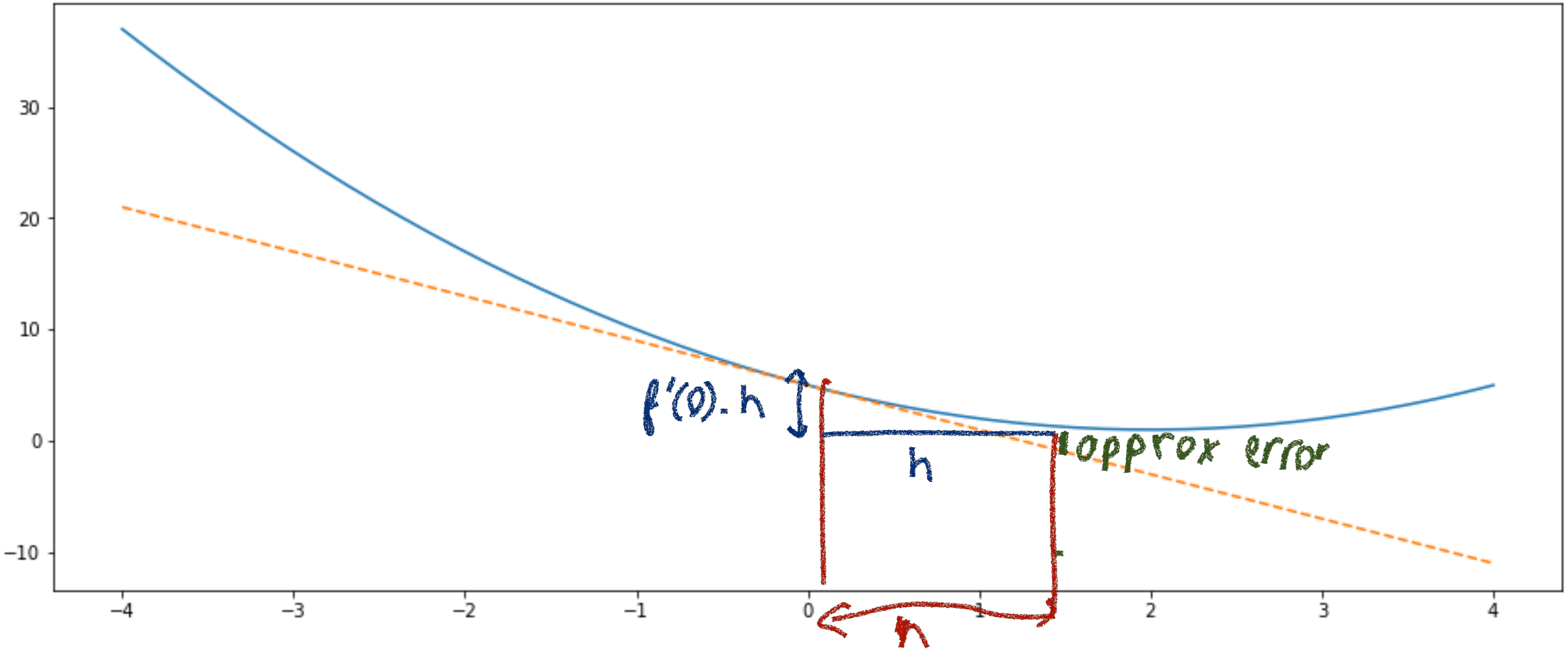
What is a derivative?

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



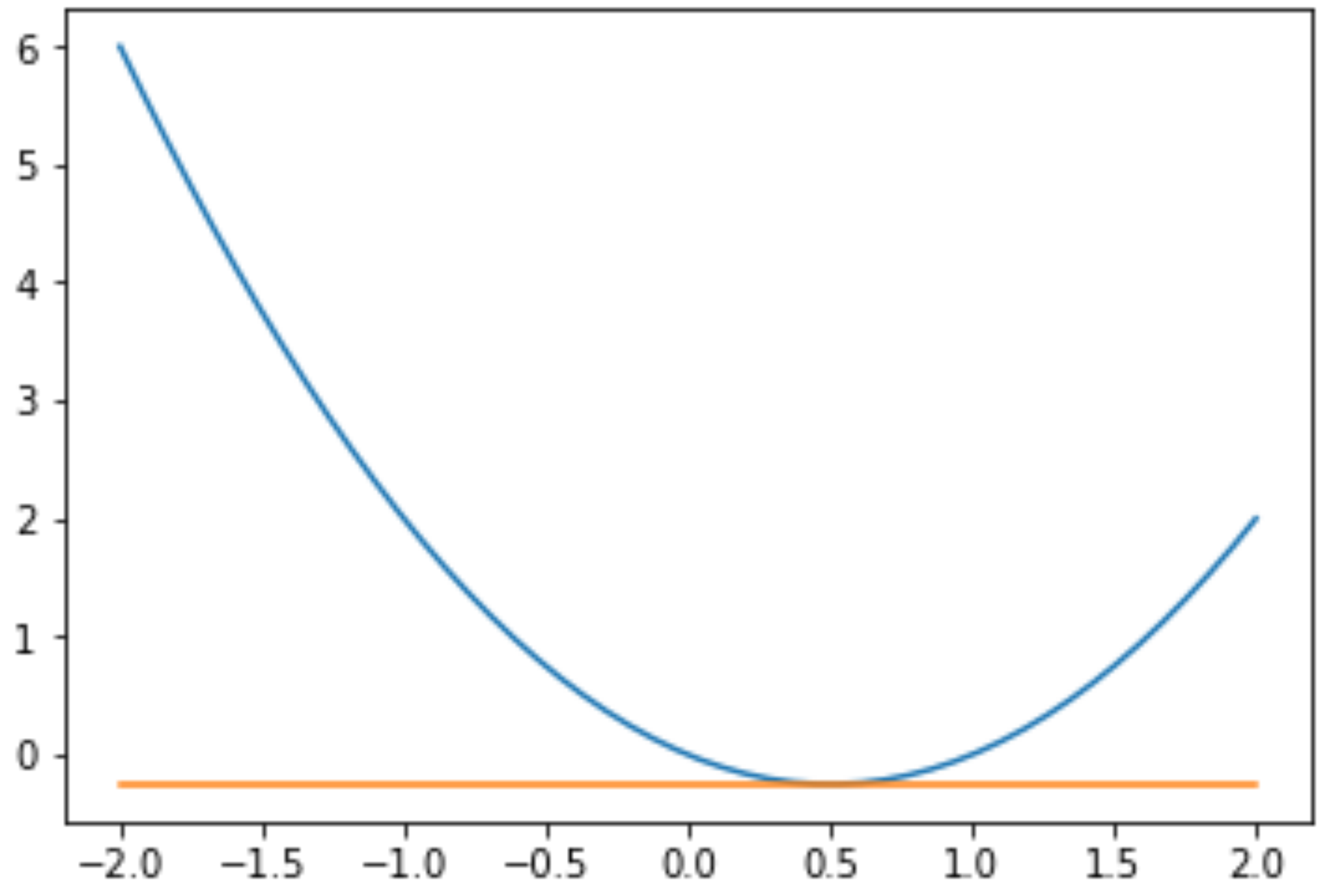
Tangent and derivative

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



Derivative and minimum

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

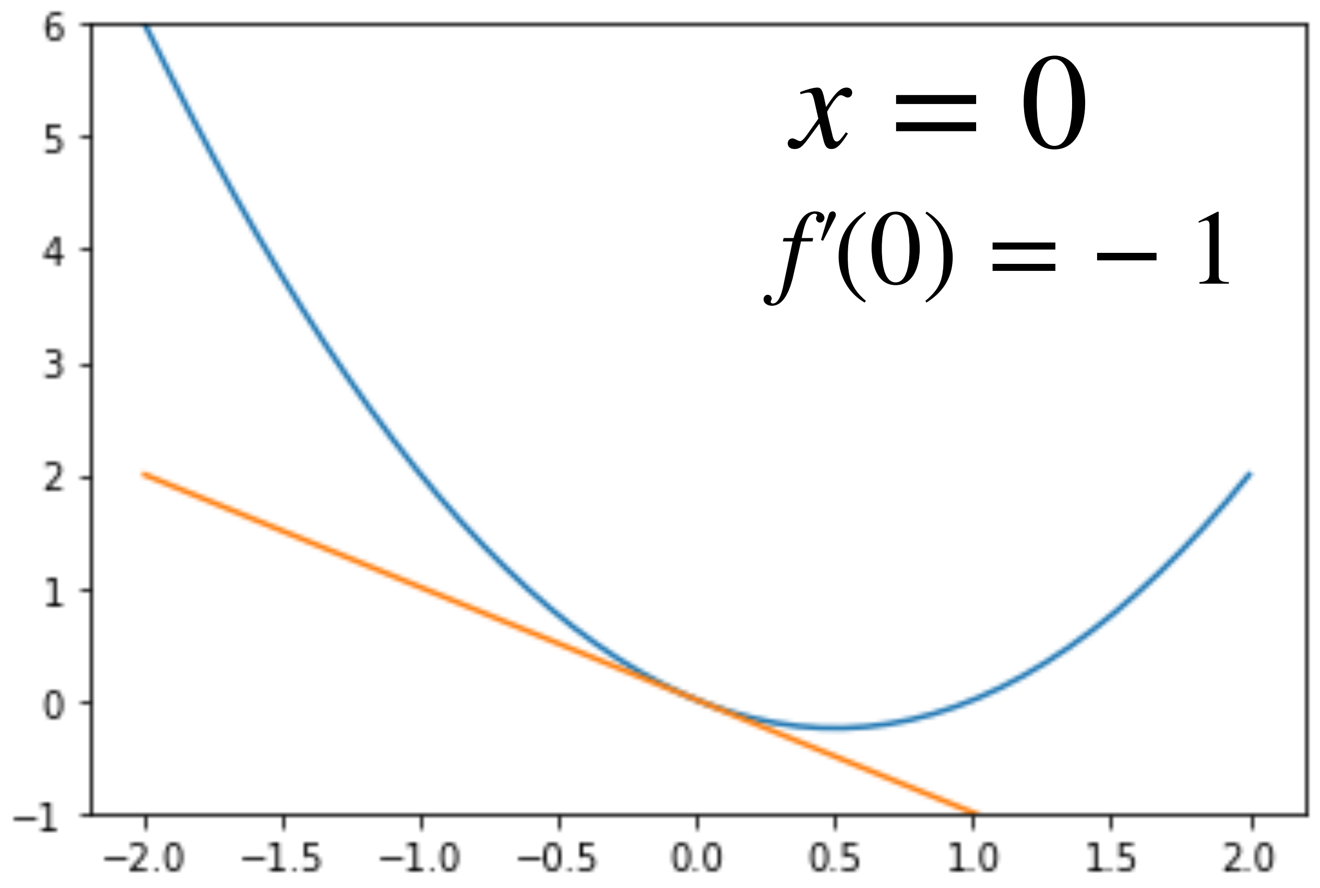


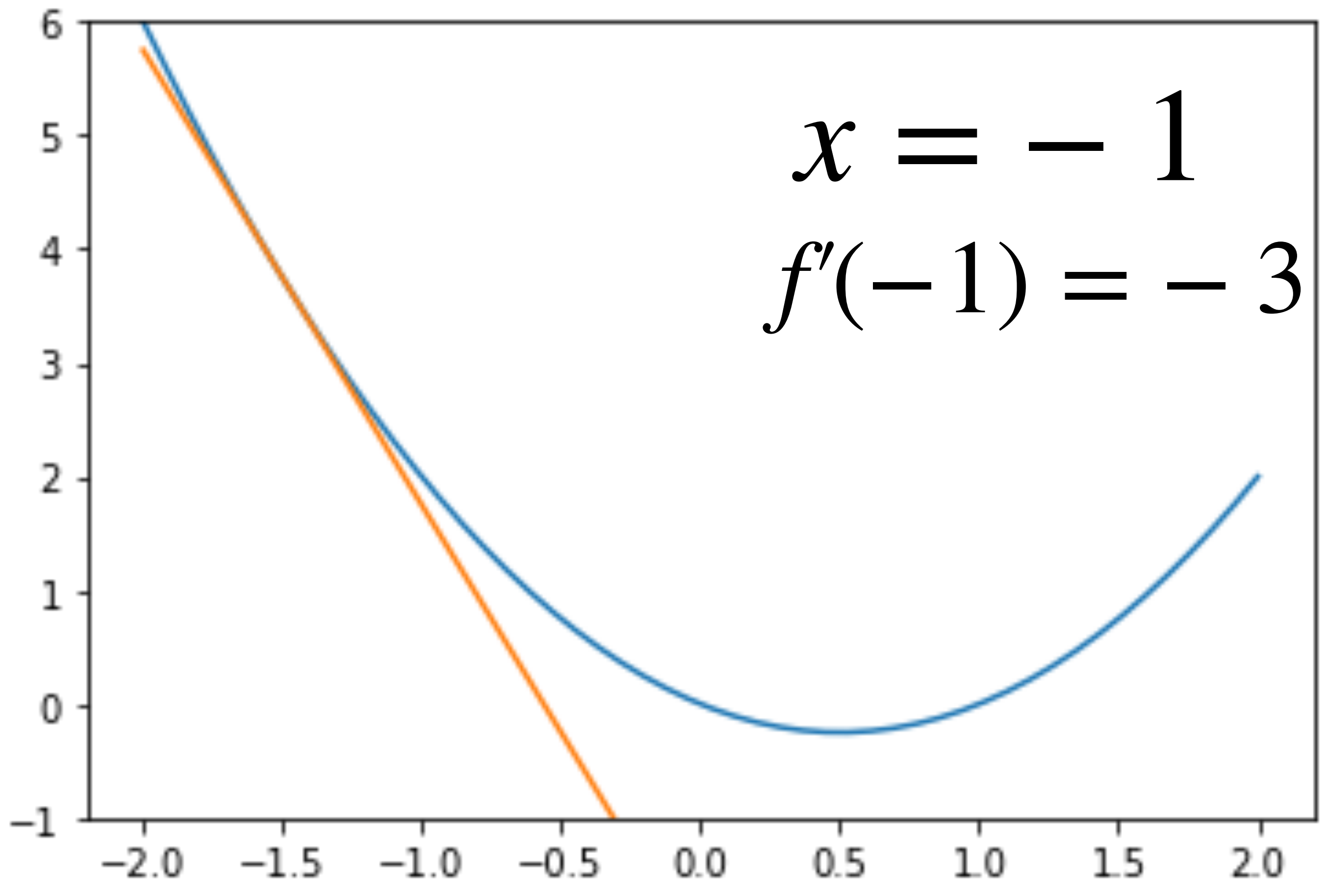
Derivative and minimum

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

Derivative and minimum

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

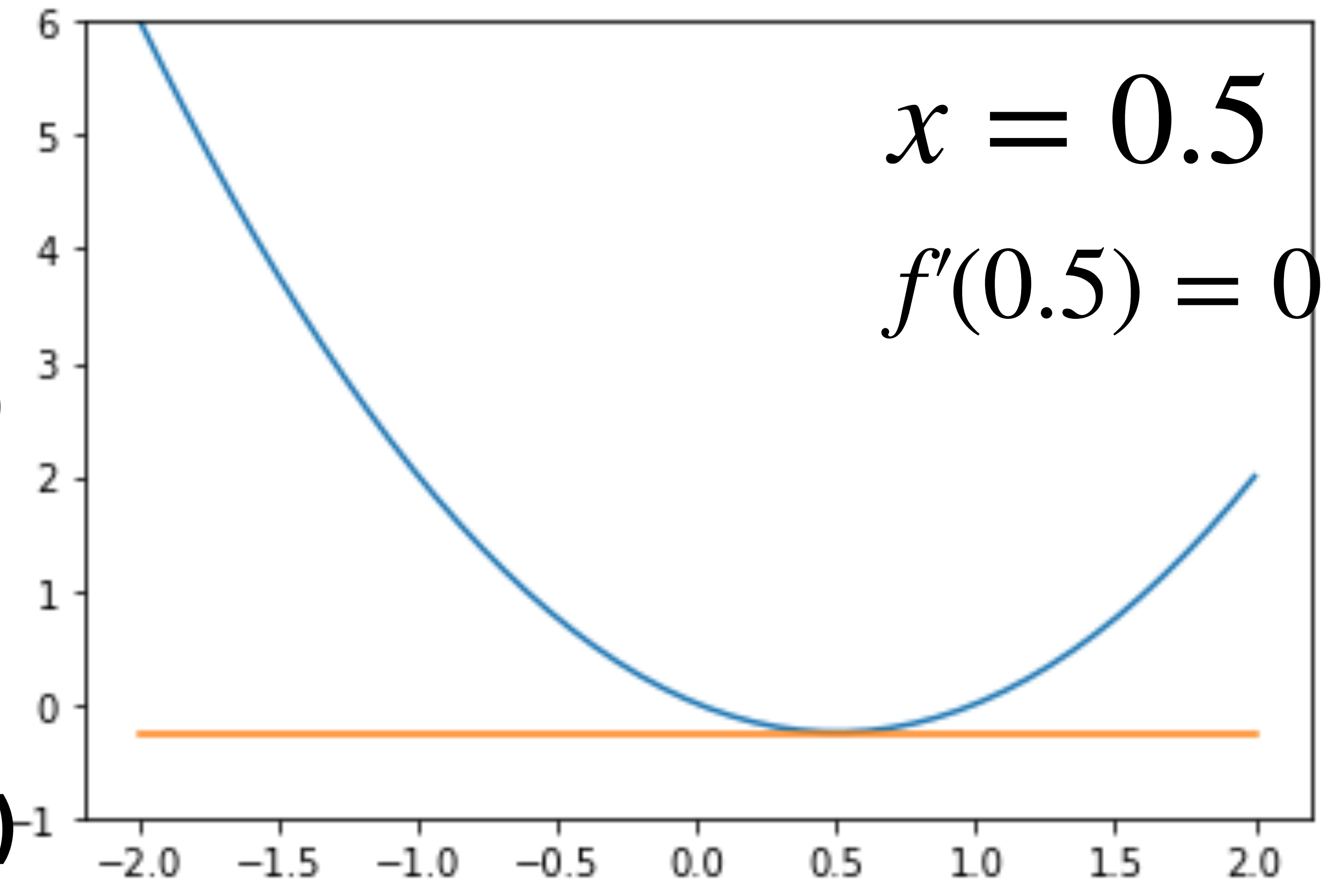


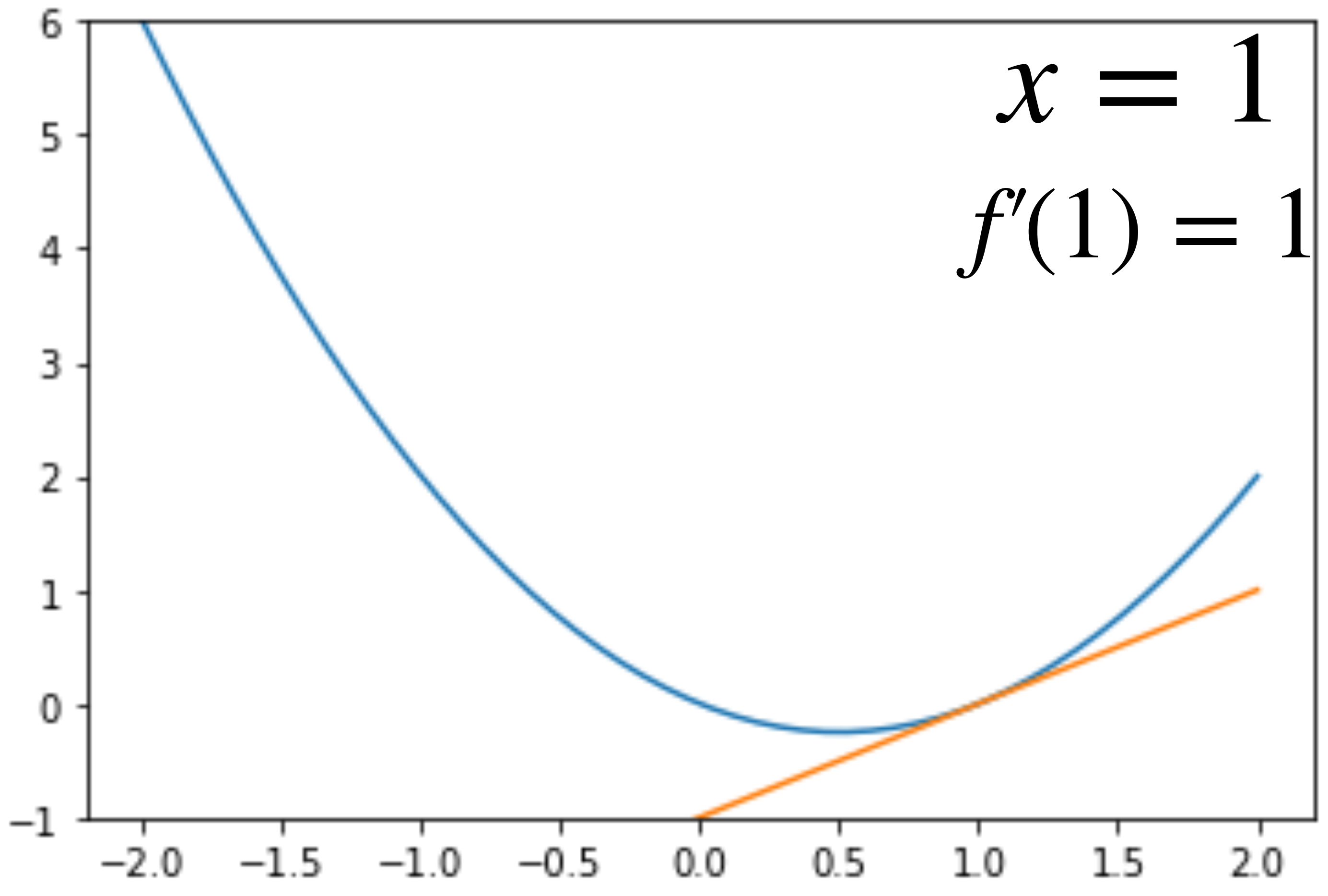


If derivative at x is negative, minimum is on the right
(we need to increase x to get closer to the minimum)

If derivative at x is positive, minimum is on the left
(we need to decrease x to get closer to the minimum)

If derivative at x is zero, x should be a minimum
(not necessarily in theory, but in our cases, it will be)





Gradient Descent Algorithm

- This suggests some procedure for finding a minimum:
  - Start at any x (eg. 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

- This suggests some procedure for finding a minimum:
  - Start at any x (eg. 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:

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

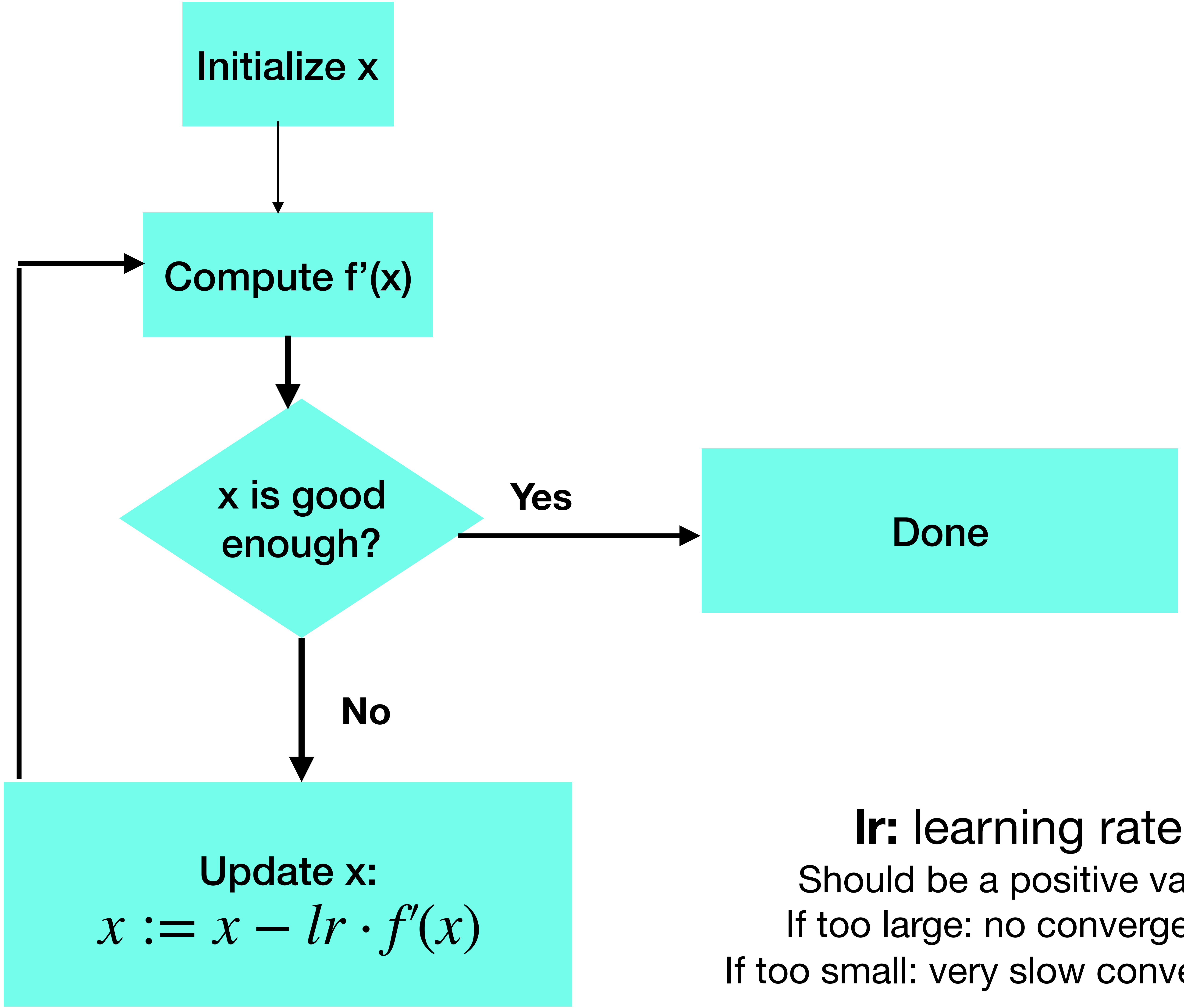
lr: learning rate

Should be a positive value

If too large: no convergence

If too small: very slow convergence

Gradient Descent algorithm



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

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

$$\arg\min_x f(x) = 0.5$$

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

- Let us try to see a bit more how it works in practice using Jupiter Notebooks
    - https://colab.research.google.com/drive/1Pdn4laPkbt-DU3w2EdidFiqdXAq63QJu
    - bit.ly/2KDOoTP

Gradient Descent Algorithm

- 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 (ie. the system learn nothing)
  - **Too small** learning rate: **slow** convergence (ie. the system takes a long time to learn)

Minimizing a function of several variables

Functions of several variables

- A function of several variables is just that: a function which has several variables

$$f : \mathbb{R}^3 \to \mathbb{R}$$

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

$$f(0,0,0) = 0$$
$$f(1,2,3) = 7$$
$$f(-1,2,2) = 11$$
$$f(0,1,1) = ?$$
$$f(2,2,0) = ?$$

- Like before, we want to find its minimum:

$$\arg\min_{x,y,z} f(x, y, z) = (0,0,0.5)$$

Parameterized functions

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

$$\downarrow$$

Fixing z: $z = 2$

$$\downarrow$$

$$f(x, y, 2) = (x - y)^2 + 4 - 2$$

Function of 2 variables: $f(x, y) = (x - y)^2 + 2$

Parameterized functions

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

$$\downarrow$$

Fixing y: $y = 2$

$$\downarrow$$

$$f(x,2,z) = (x - 2)^2 + z^2 - z$$

Function of 2 variables: $f(x, z) = (x - 2)^2 + z^2 - z$

Parameterized functions

- By fixing one of the variable, we can obtain a function with one less variable

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

$$\downarrow$$

Fixing y and z: $y = 2$ $z = 3$

$$\downarrow$$

$$f(x,2,3) = (x - 2)^2 + 9 - 3$$

Function of 1 variable: $f(x) = (x - 2)^2 + 6$

Parameterized functions

- Therefore, in this case, variables **y** and **z** can be used to describe a "family" of functions.

- We say they parameterize the

Function of 3 variables: $f(x, y, z) = (x - y)^2 + z^2 - z$

$$\downarrow$$

Fixing y and z: $y = 2$ $\quad z = 3$

$$\downarrow$$

$$f(x,2,3) = (x - 2)^2 + 9 - 3$$

Function of 1 variable: $f(x) = (x - 2)^2 + 6$

For each value of y and z, we have one function of one variable

Parameterized functions

- Therefore, in this case, variables **y** and **z** can be used to describe a “family” of functions
- In such a case, we will say that f is a function *parameterized* by **y** and **z**
- And we note the parameters separately, as subscripts

Function of 3 variables: $f_{y,z}(x) = (x-y)^2 + z^2 - z$

$$\downarrow$$

Fixing y and z: $y = 2$ $\quad z = 3$

$$\downarrow$$

$$f_{2,3}(x) = (x-2)^2 + 9 - 3$$

Function of 1 variable: $f_{2,3}(x) = (x-2)^2 + 6$

For each value of y and z, we have one function of one variable

$$f_{0,0}(x) = x^2$$

$$f_{0,2}(x) = x^2 + 2$$

Partial derivatives

- What is the equivalent of our “high school” derivatives when we have several variables?

- One part of the answer is ***partial derivatives***

- Partial derivatives are computed by <u>choosing one variable</u> and <u>fixing the others</u>

- In other words, we see the function of several variables as a <u>parameterized function of one variable</u>

Partial derivatives

- What is the equivalent of our "high school" derivatives when we have several variables?
- One part of the answer is ***partial derivatives***
- Partial derivatives are computed by <u>choosing one variable</u> and <u>fixing the others</u>
- In other words, we see the function of several variables as a parameterized function of one variable
- Indeed, if we choose y, and fix x and z, we can see f(x, y, z) as a function of one variable and compute its derivative

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

- $\dfrac{\partial f}{\partial x} = 2(x - y)$ $\qquad \dfrac{\partial f}{\partial y} = 2(y - x)$ $\qquad \dfrac{\partial f}{\partial z} = 2z - 1$

Partial derivatives

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

Fix other variables

$$f_{y,z}(x) = (x - y)^2 + z^2 - z \qquad f_{x,z}(y) = (x - y)^2 + z^2 - z \qquad f_{x,y}(z) = (x - y)^2 + z^2 - z$$

Compute derivative

$$f'_{y,z}(x) = 2(x - y) \qquad f'_{x,z}(y) = 2(y - x) \qquad f'_{x,y}(z) = 2z - 1$$

Partial derivatives

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

Fix other variables

$$f_{y,z}(x) = (x - y)^2 + z^2 - z \qquad f_{x,z}(y) = (x - y)^2 + z^2 - z \qquad f_{x,y}(z) = (x - y)^2 + z^2 - z$$

Compute derivative

$$f'_{y,z}(x) = 2(x - y) \qquad f'_{x,z}(y) = 2(y - x) \qquad f'_{x,y}(z) = 2z - 1$$

In practice, we use this notation for partial derivatives:

$$\frac{\partial f}{\partial x} = 2(x - y) \qquad \frac{\partial f}{\partial y} = 2(y - x) \qquad \frac{\partial f}{\partial y} = 2z - 1$$

Computing the partial derivatives

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

$$\frac{\partial f}{\partial x} =$$

$$\frac{\partial f}{\partial y} =$$

$$\frac{\partial f}{\partial z} =$$

Partial derivatives

- Exercise: Compute the partial derivatives

$$f(x, y, z) = xyz - z^2 - y^2$$

$$f(x, y, z) = e^{x+y} - log(z)$$

Partial derivatives

- Exercise: Compute the partial derivatives

$$f(x, y, z) = xyz - z^2 - y^2$$

Choice A:

$$\frac{\partial f}{\partial x} = yz$$

$$\frac{\partial f}{\partial y} = xz - 2y$$

$$\frac{\partial f}{\partial z} = xy - 2z$$

Choice B:

$$\frac{\partial f}{\partial x} = x - z^2 - y^2$$

$$\frac{\partial f}{\partial y} = y - 2y - z^2$$

$$\frac{\partial f}{\partial z} = z - 2z - y^2$$

Partial derivatives

- Exercise: Compute the partial derivatives

$$f(x, y, z) = e^{x+y} - log(z)$$

Choice A:

$$\frac{\partial f}{\partial x} = e^{x+y}$$

$$\frac{\partial f}{\partial y} = e^{x+y}$$

$$\frac{\partial f}{\partial z} = -\frac{1}{z}$$

Choice B:

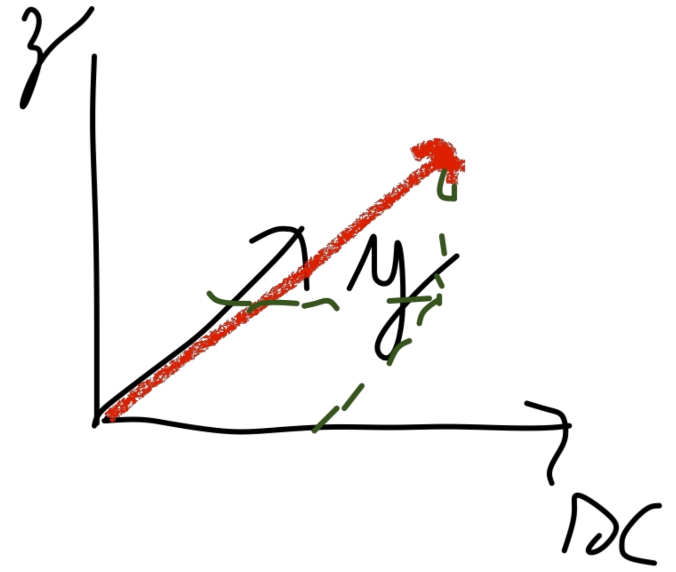
$$\frac{\partial f}{\partial x} = e^{x+y} - \log(z)$$

$$\frac{\partial f}{\partial y} = e^{x+y} - \log(z)$$

$$\frac{\partial f}{\partial z} = e^{x+y} - \frac{1}{z}$$

Vectors

- What are vectors?
- You probably have used vectors in Physics classes to represent force and speed
  - 3-dimensional vectors: [2.3, 4.5, -1]



- In Machine Learning, we also use them a lot
- Except that they can have more than 3 dimensions
  - 5-dimensional vector: [-1, 3, 4.1 ,5.2, 4]
  - We often note the set of all n-dimensional vectors $\mathbb{R}^n$

$$[1, 2.1, 4.1, -1, -1] \in \mathbb{R}^5$$

Vectors (Continued)

- For now, we only need to know the following about vectors:
    - A n-dimensional Vector is a list of n numbers
    - We can add 2 vectors (if they have the same dimension)

$$[2.1, 3.4, 1.1, 3.2] + [-1, 2.1, 3.1, -2] = [1.1, 5.5, 4.2, 1.2]$$

$$[2.1, 3.4] + [-1, 2.1, 3.1, -2] =$$ ✗

-
    - We can multiply a vector by a number

$$0.5 \times [2, 3, -1, -2] = [1, 1.5, -1.5, -1]$$

Vectors(Continued)

- We will usually denote a vector by a letter with an arrow on it: $\overrightarrow{x}$
- We denote the i^{th} component of $\overrightarrow{x}$ by x_i
- If $\overrightarrow{x} = [1,2.2, -1,4]$
  - Then we have $x_0=1$, $x_1=2.2$, $x_2=-1$, $x_3=4$

Vectors: Exercise

$$\overrightarrow{x} = [1,5,-2,0.5]$$

$$\overrightarrow{y} = [2,2,10,10]$$

$$\vec{z} = [3,-3,0]$$

- Dimensions of $\overrightarrow{x}, \overrightarrow{y}, \vec{z}$?
- Values of x_1, y_2, z_0, y_0?
- Compute:

$$\overrightarrow{x} + \overrightarrow{y}$$

$$\overrightarrow{x} + 0.5 \times \overrightarrow{y}$$

$$\overrightarrow{y} + \vec{z}$$

Vectors and Multivariate functions

- For now, we have represented the variables of a multivariate function with the letters x, y, z as in: $$f(x, y, z) = (x - y)^2 + z^2 - z$$
- In practice, we can have any number of variables. So it is more convenient to use:
  - x_0 (instead of x) , x_1 (instead of y), x_2 (instead of z), x_3 .. x_n (if we need more than 3 variables) $$f(x_0, x_1, x_2) = (x_0 - x_1)^2 + x_2^2 - x_2$$
- We can also **use a vectorial notation** to represent all of the variables as one vector variable:

$$\overrightarrow{x} = [x_0, x_1, x_2] \qquad f(\overrightarrow{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

- So, keep in mind that the 3 following expressions actually refer to the same function:

$$f(x, y, z) = (x - y)^2 + z^2 - z$$

$$f(x_0, x_1, x_2) = (x_0 - x_1)^2 + x_2^2 - x_2$$

-

$$f(\overrightarrow{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

Gradient

- The partial derivatives become the component of a vector we call the ***gradient***

$$grad \cdot f(x, y, z) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]$$

- For example: $f(x, y, z) = (x - y)^2 + z^2 - z$
- $grad \cdot f(x, y, z) = [2(x - y), 2(y - x), 2z - 1]$

Gradient

$$grad \cdot f(x, y, z) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}]$$

- In this case, the function has 3 variables. Therefore the gradient is a vector of size 3
- If the gradient has n variables, it is a vector of size n
- More precisely, the gradient of f is itself a function that return a vector

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

$$f(x_1, x_2, \ldots, x_n)$$

$$grad \cdot f : \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$grad \cdot f(x_1, x_2, \ldots, x_n) = [g_1, \ldots, g_n]$$

Interpreting the gradient

- At a given point, the gradient is the direction for which the value of the function increase fastest

- Therefore, in general, it points in the direction **opposite to the minimum**

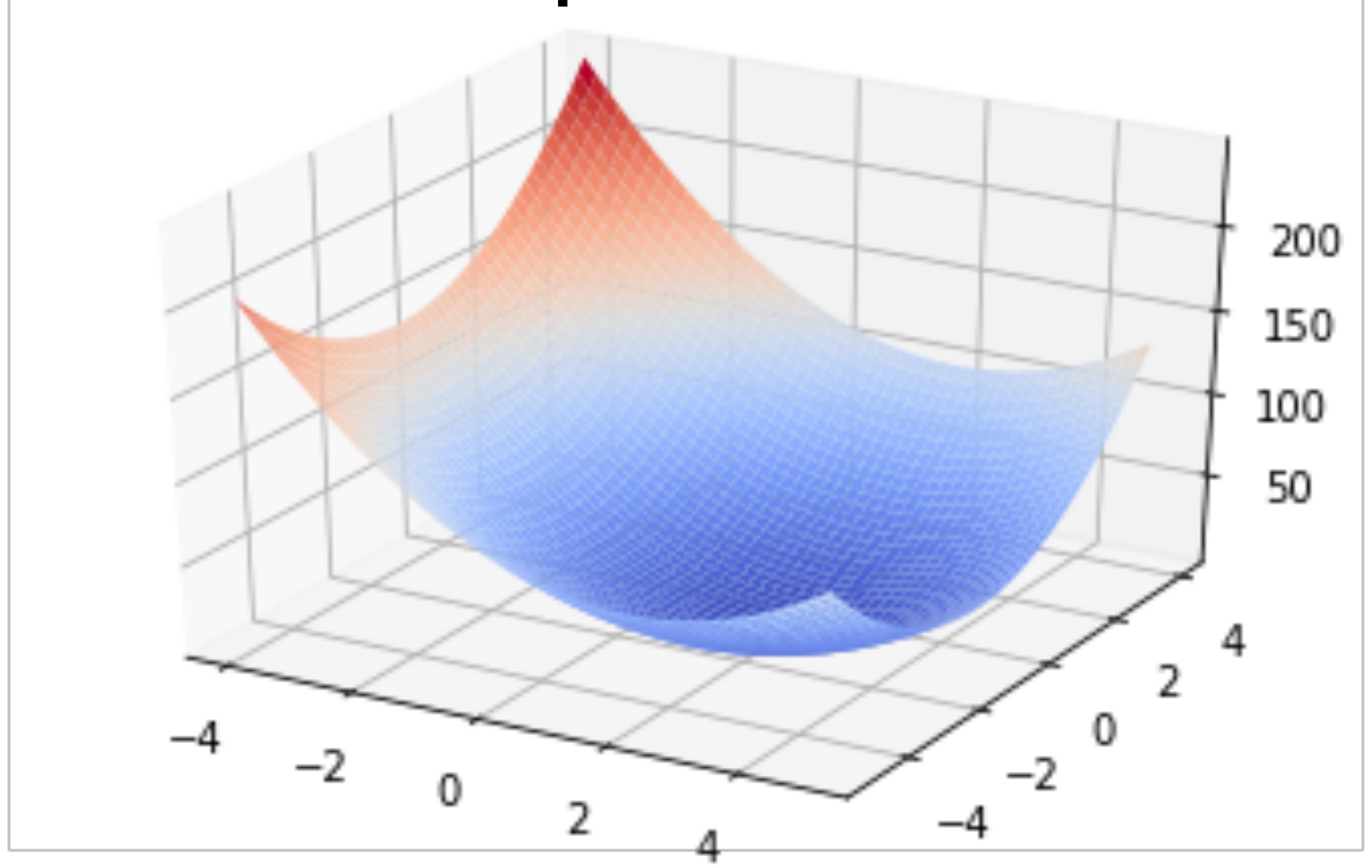
$$f(x, y) = 4(x-2)^2 + 4(y+1)^2 - 0.1xy$$

$$grad \cdot f(x, y) = [8(x-2) - 0.1y, 8(y+1) - 0.1x]$$

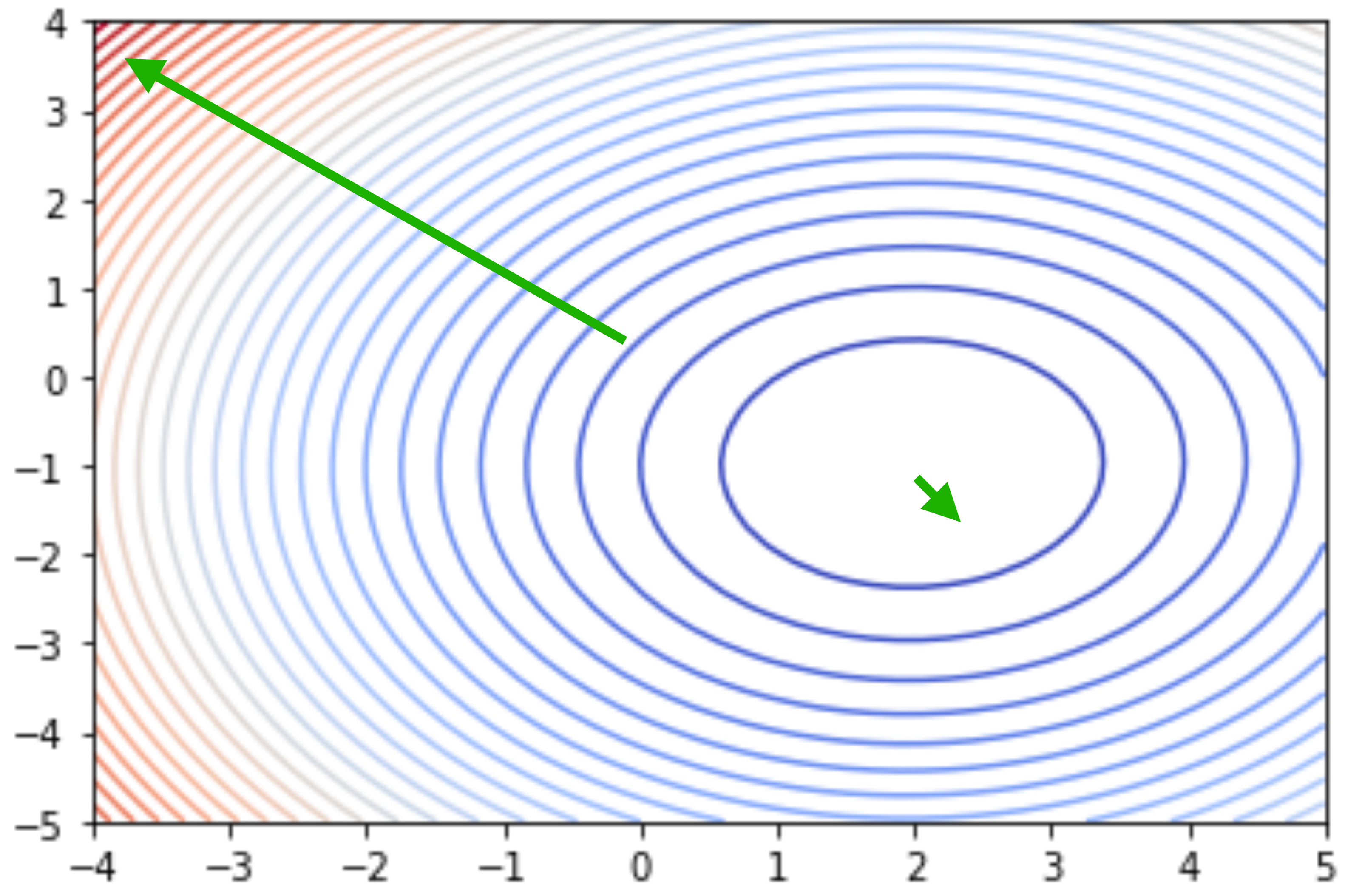
$$grad \cdot f(0,0) = [-16, 16]$$

$$grad \cdot f(2, -1) = [0.1, -0.2]$$

3D plot



Contour Plot



Gradient Descent

- Because we know that the gradient point in a direction opposite to the minimum, we can use the same idea as in the case of one variable

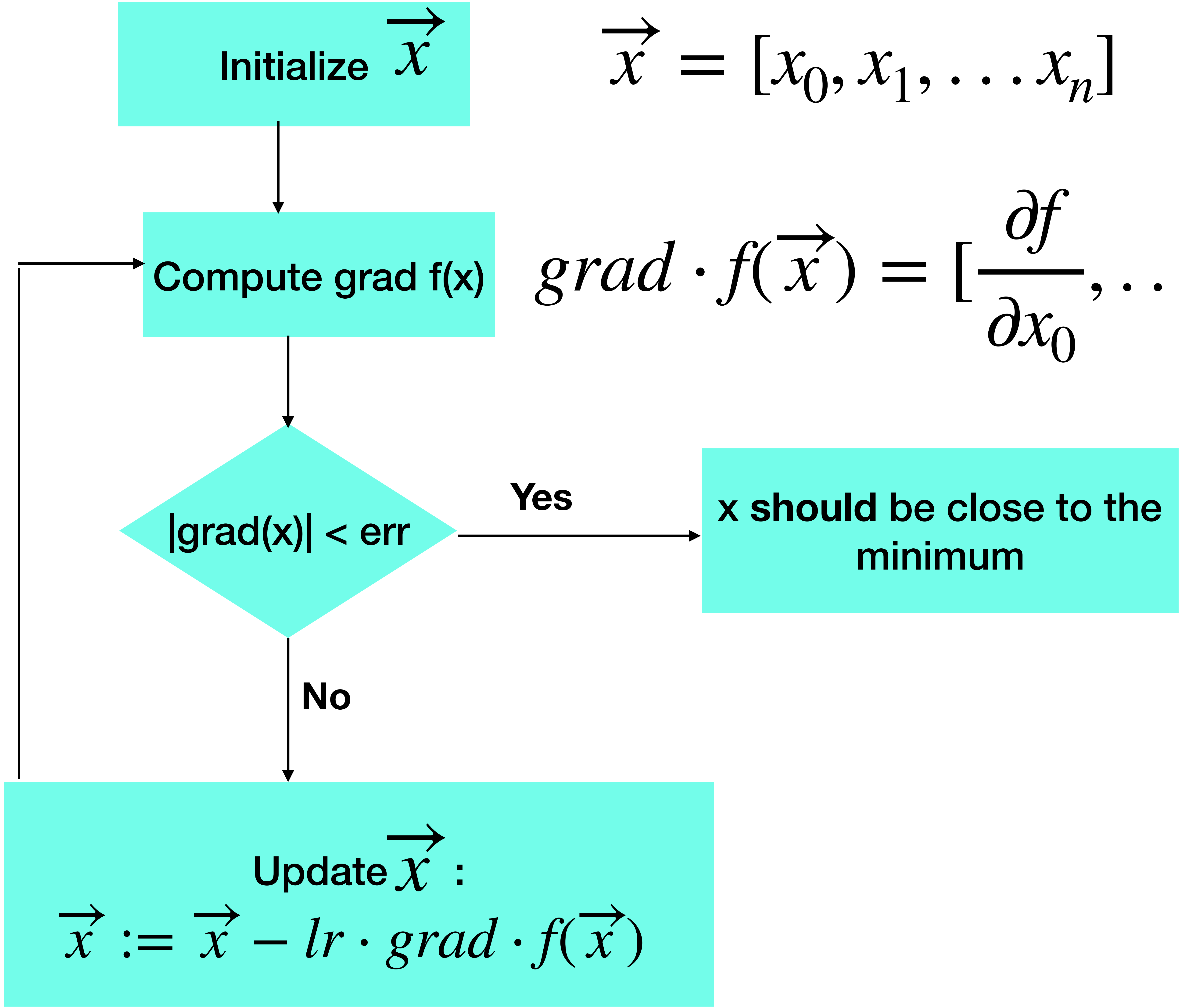
One variable:

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

Multiple variables:

$$\overrightarrow{x} := \overrightarrow{x} - lr \cdot grad \cdot f(\overrightarrow{x})$$

Gradient Descent algorithm



$$\vec{x} = [x_0, x_1, \ldots x_n]$$

$$grad \cdot f(\vec{x}) = [\frac{\partial f}{\partial x_0}, \ldots \frac{\partial f}{\partial x_n}]$$

$$f(\vec{x}) = (x_0 - x_1)^2 + x_2^2 - x_2$$

$$grad \cdot f(\vec{x}) = [2(x_0 - x_1), 2(x_1 - x_0), 2x_2 - 1]$$

$$lr = 0.2$$

$$\vec{x} = (0,1,0)$$

$$grad \cdot f(\vec{x}) = [-2,2,-1]$$

$$\vec{x} = (0.4,0.6,0.2)$$

$$grad \cdot f(\vec{x}) = [-0.4,0.4,-0.6]$$

$$\vec{x} = (0.41,0.43,0.51)$$

$$grad \cdot f(\vec{x}) = [-0.04,0.04,0.01]$$

- Let us check gradient descent in practice with some notebook

- On google colab: http://bit.ly/2vertEi

- (or full url: https://colab.research.google.com/drive/16MvnlY0TH8HTEiDCgRoWLZOqzGPu4np8)

-

What is the equivalent of second derivative for multivariate functions?

- It is the Hessian Matrix:

$$\begin{vmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial x \partial z} & \frac{\partial^2 f}{\partial y \partial z} & \frac{\partial^2 f}{\partial z^2} \end{vmatrix}$$

- But <u>thankfully, we will not need to use</u> it
- But <u>for your information</u>, this would be the equivalent of the "High School" minimization when we have several variables:
-

To minimize f(x, y, z):

1. Compute **gradient** of f(x, y , z)
2. Compute **hessian** of f(x)
3. Find x, y, z such that grad f(x,y,z) = 0
4. If hessian of f(x,y,z) is **definite positive** then (x,y,z) is a **local minimum** of f

Gradient Descent Algorithm

- You can see that, in the case of the gardient descent, the algorithm is the same for univariate functions and multivariate functions
- It is a simple algorithm, but it scales very well
- There exists many variations of it:
  - Gradient Descent with momentum
  - Stochastic Gradient Descent
  - Adagrad, Adadelta, Adam, …

Gradient Descent with Momentum

- Compute a “gradient with momentum” at each iteration:
-

$$gm_t = 0.6grad \cdot f(\overrightarrow{x}) + 0.4gm_{t-1}$$

Update $\overrightarrow{x}$:

$$\overrightarrow{x} := \overrightarrow{x} - lr \cdot gm_t$$

Stochastic Gradient Descent

- What happens if the gradient is noisy?
- That is, we can only compute a value that is equal to the true gradient "on average"?
    - A bit like if you are drunk and trying to get home

Stochastic Gradient Descent

- What happens if the gradient is noisy?
- That is, we can only compute a value that is equal to the true gradient “on average”?
  - A bit like if you are drunk and trying to get home
- It turns out it works.
  - But you have to decrease your learning rate over time to stabilize $lr = \frac{lr_0}{\sqrt{(}t+1)}$
  - Convergence will be slower
- Very interesting because a noisy gradient can be million times faster to compute than a “true” gradient

Optimization libraries

- You can also minimize a function by using a specialized library
- It gives you access to more sophisticated minimization algorithms
- However these more sophisticated algorithms do not scale as well as Gradient Descent
  - Which is one Gradient Descent and its variants are still the main tool for large scale Machine Learning (In particular, Deep Learning)

```
In [103]: scipy.optimize.minimize(f_numpy, np.array([0,0]), jac=grad_f_numpy, method="L-BFGS-B")

Out[103]:       fun: 0.1969057665260197
           hess_inv: <2x2 LbfgsInvHessProduct with dtype=float64>
                jac: array([-3.88578059e-16,  2.77555756e-17])
            message: b'CONVERGENCE: NORM_OF_PROJECTED_GRADIENT_<=_PGTOL'
               nfev: 5
                nit: 4
             status: 0
            success: True
                  x: array([ 1.9878106 , -0.97515237])
```