Machine Learning

Topic 7. Lecture 7

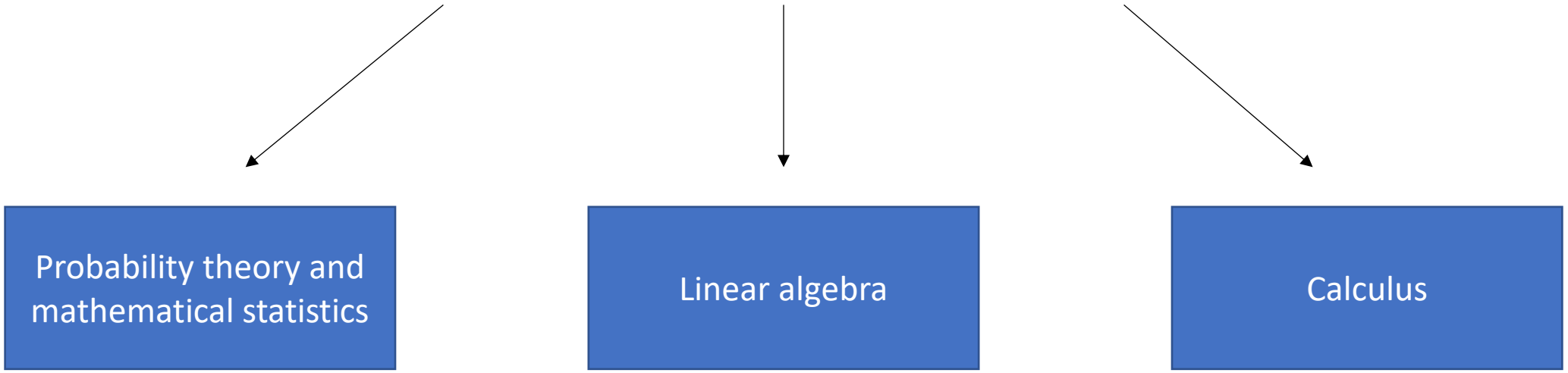
Math for machine learning. Calculus

Yury Sanochkin

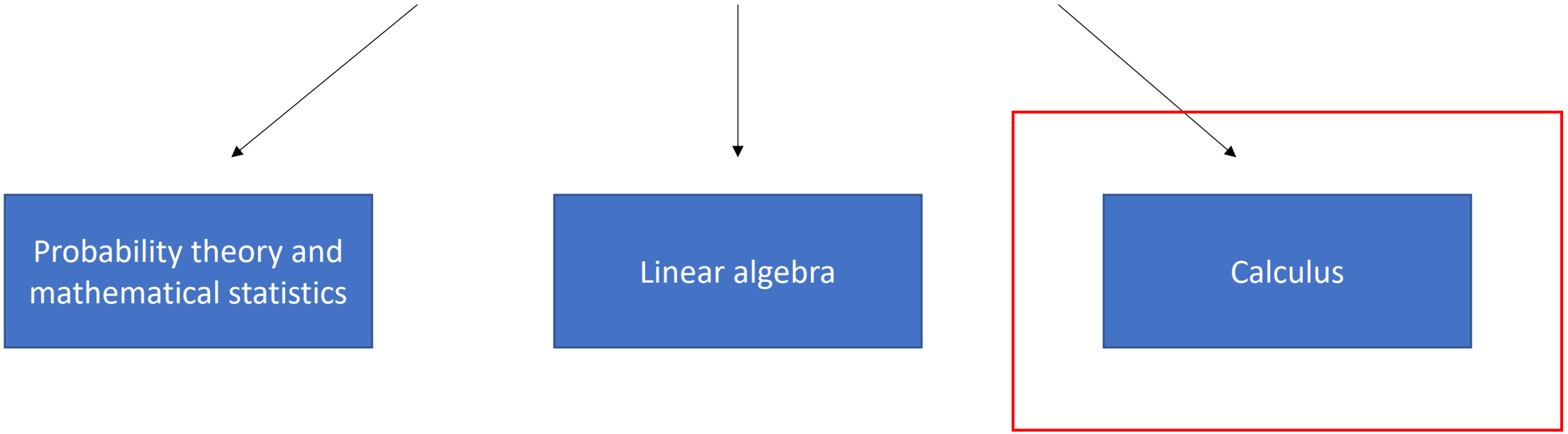
ysanochkin@hse.ru

NRU HSE, 2025

Math for machine learning



Math for machine learning



Today, as we agreed, we will talk about this!

Calculus. Motivation

Calculus. Motivation

- Let's follow the pattern we are already familiar with and first discuss why calculus is so important in the context of machine learning tasks.
- How can you answer this question yourself: what is this section about and what, in your opinion, is its significance?

Calculus. Motivation

- Differential optimization – the fiery engine of machine learning!

Calculus. Motivation

- Differential optimization – the fiery engine of machine learning!
- Is that too bold a statement?

Calculus. Motivation

- Differential optimization – the fiery engine of machine learning!
- Is that too bold a statement?
- Not at all.
- We will progress step by step in calculus and gradually understand why, in reality, this is absolutely the case.

Calculus. Motivation

- Let's recall: what is the main task of classical machine learning?

Calculus. Motivation

- Let's recall: what is the main task of classical machine learning?
- The main task is to reconstruct the hidden (unknown) dependency in the data as accurately as possible.

Calculus. Motivation

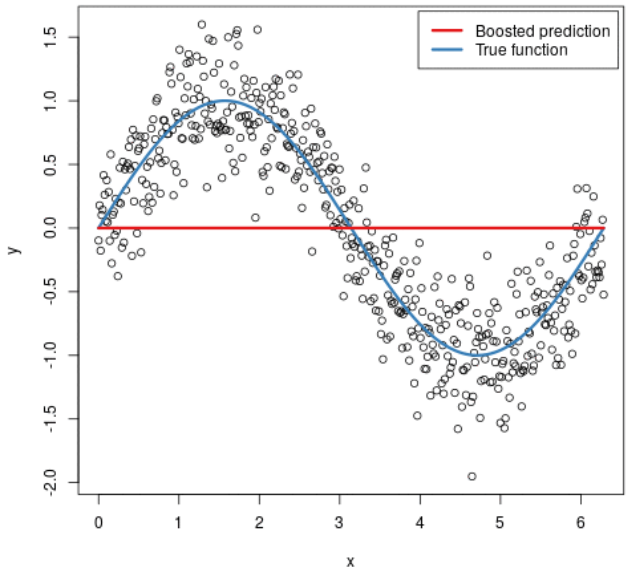
- Let's recall: what is the main task of classical machine learning?
- The main task is to reconstruct the hidden (unknown) dependency in the data as accurately as possible.
- This dependency can be functional; have a stochastic nature; represent a cluster structure generating the data distribution; or be something else entirely. Essentially, that is not so important here.
- The most important thing is that machine learning is, in any case, about identifying and detecting such a dependency.

Calculus. Motivation

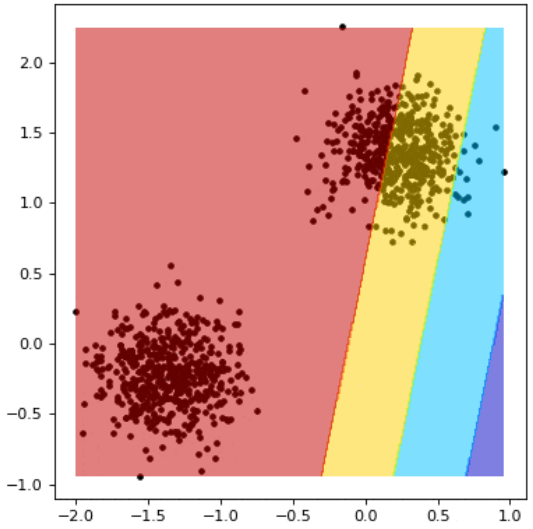
- So once again, let's conceptually recall how this all works:

Calculus. Motivation

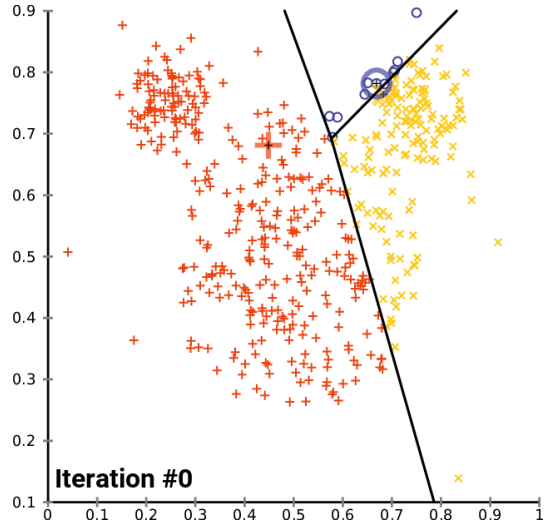
- So once again, let's conceptually recall how this all works:



Solving the regression problem using gradient boosting.



Solving the classification problem using the support vector machine method.



Solving the clustering problem using the KMeans method.

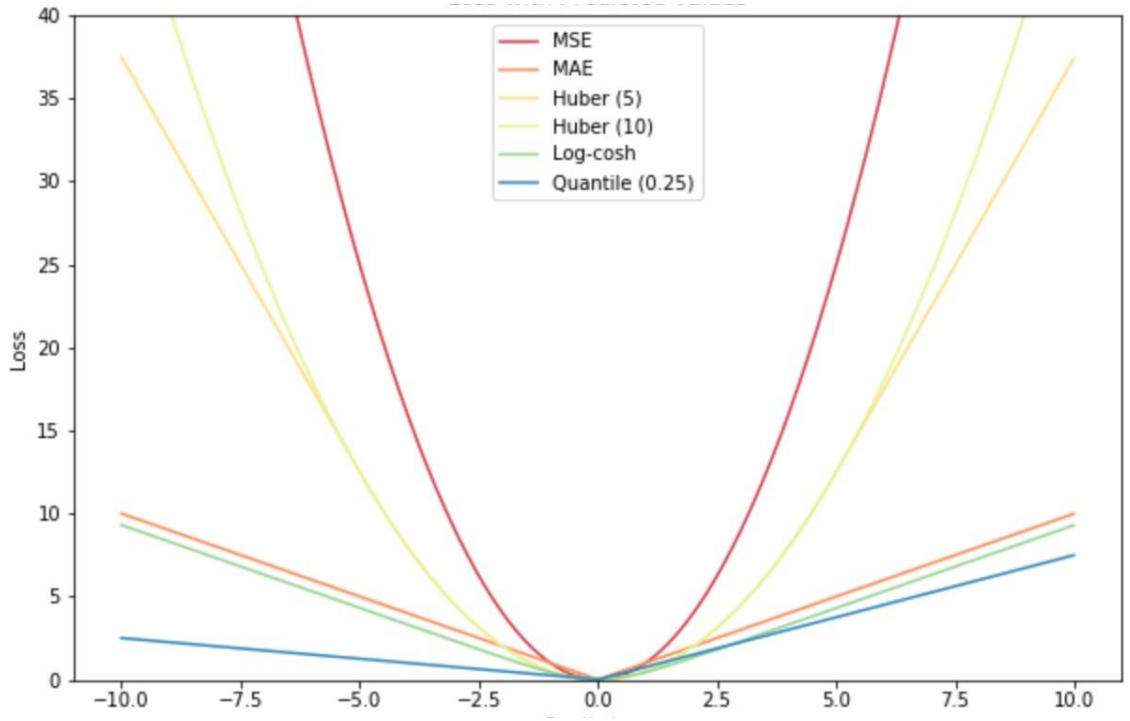
Calculus. Motivation

- But there is a problem that immediately arises here, and it's quite understandable: how can we determine whether an algorithm is performing well in its task or not so well?
- And in general — what is considered good, and what is bad in the context of ML? How do we explain this to a computer?

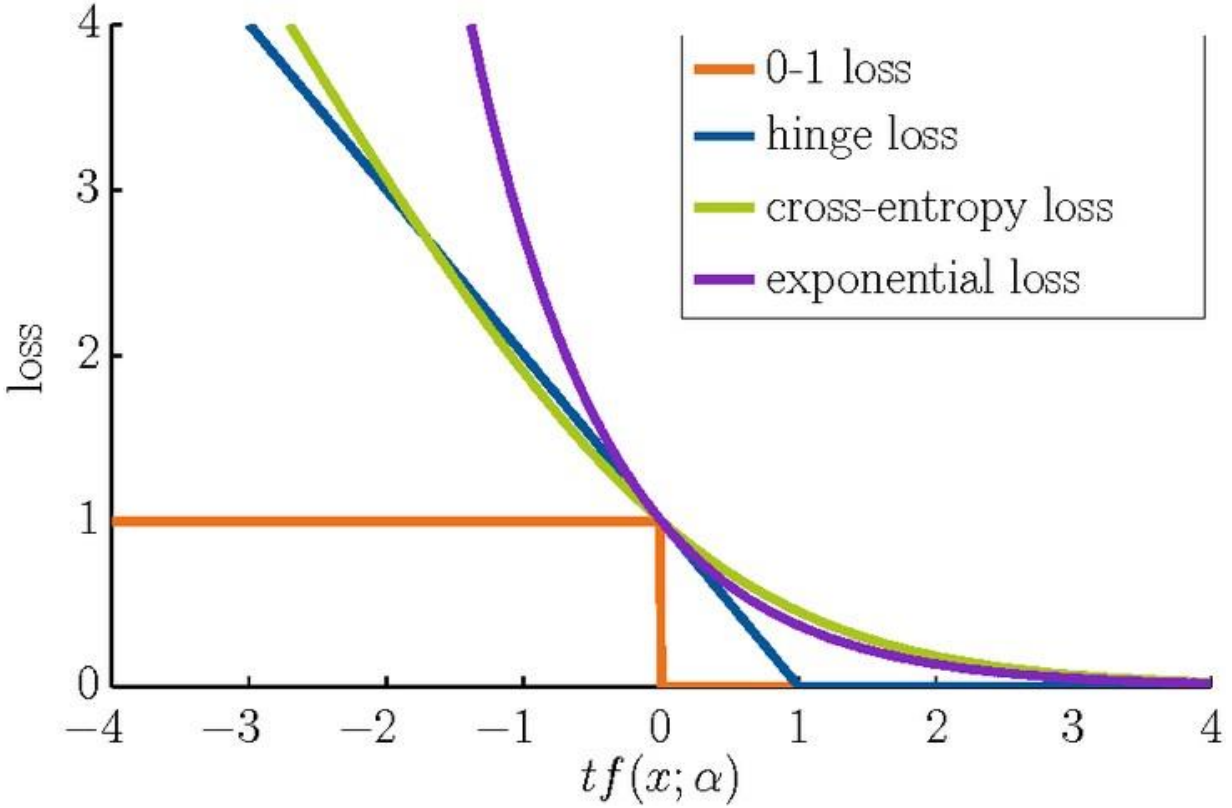
Calculus. Motivation

- But there is a problem that immediately arises here, and it's quite understandable: how can we determine whether an algorithm is performing well in its task or not so well?
- And in general — what is considered good, and what is bad in the context of ML? How do we explain this to a computer?
- It turns out that during training, our algorithm almost always tries to minimize a certain function — the so-called error function — and it does this on the data it is actually trained on!

Error Function



Typical error functions in regression tasks



Typical error functions in binary classification tasks

Error Function

- Without delving super deeply into the various types of error functions for now, it's important for us to understand the main point: regardless of the machine learning task at hand, virtually any algorithm solving it will be minimizing some functional, either explicitly or implicitly.

Error Function

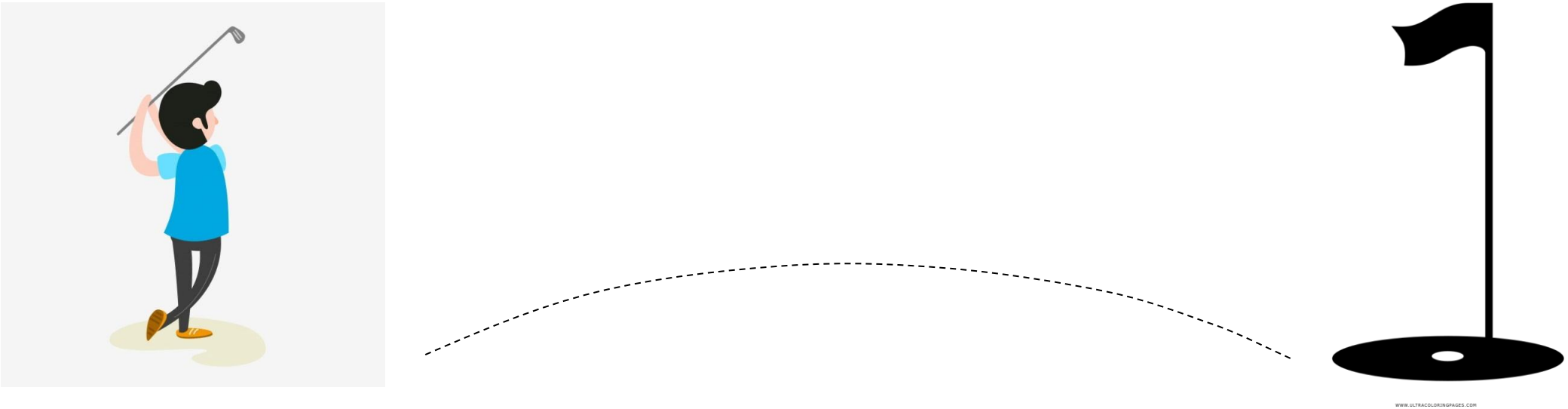
- Without delving super deeply into the various types of error functions for now, it's important for us to understand the main point: regardless of the machine learning task at hand, virtually any algorithm solving it will be minimizing some functional, either explicitly or implicitly.
- In the context of ML, this minimization occurs through the adjustment of the algorithm's parameters. Moreover, the number of these parameters can vary widely: from one, two, or a couple of dozen, to millions or even billions (as is the case with neural networks like GPT and similar).

Error Function

- A clear example of optimization:

Error Function

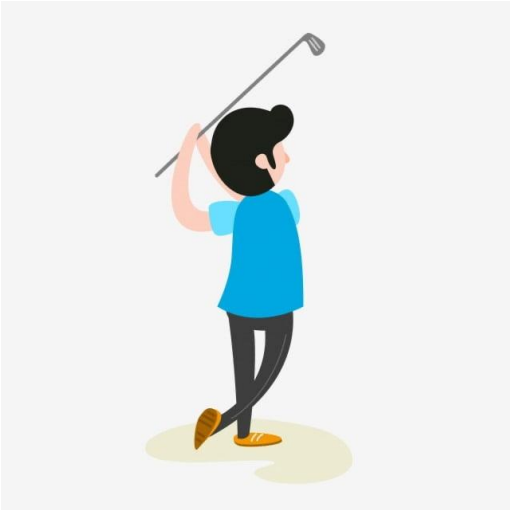
- A clear example of optimization:

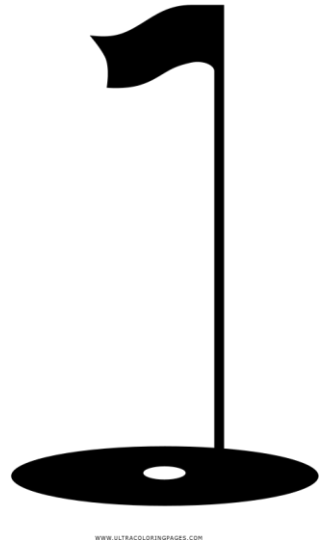


Error Function

- A clear example of optimization:

Player - ML-algorithm





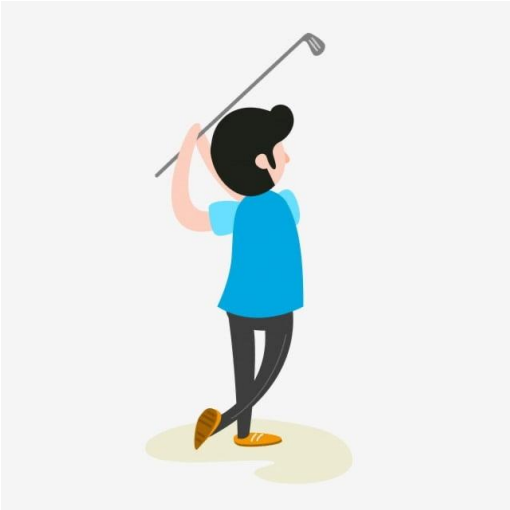
The parameter to be optimized is the impact force

The error function is how far the player missed the hole

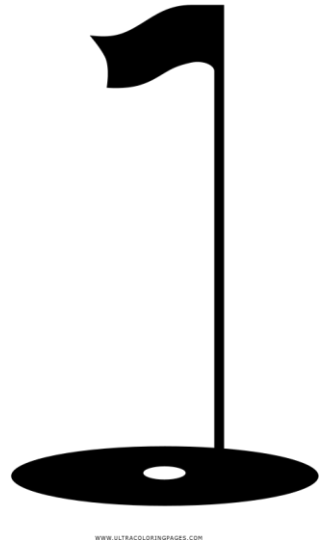
Error Function

- A clear example of optimization:

Player - ML-algorithm



The optimization algorithm is us!



The parameter to be optimized is the impact force

The error function is how far the player missed the hole

Error Function

- General scheme of training ML algorithms:

Error Function

- General scheme of training ML algorithms:
- While the error is decreasing:
  - Make predictions on the training dataset.

Error Function

- General scheme of training ML algorithms:
- While the error is decreasing:
  - Make predictions on the training dataset.
  - Calculate the error function.

Error Function

- General scheme of training ML algorithms:
- While the error is decreasing:
  - Make predictions on the training dataset.
  - Calculate the error function.
  - Update the model parameters to minimize the error function.

Error Function

- Alright. Let's assume we understand how to calculate the error function and determine if it changes—that's clear.

Error Function

- Alright. Let's assume we understand how to calculate the error function and determine if it changes—that's clear.
- But how do we change the parameters so that the algorithm makes fewer mistakes?

Error Function

- Alright. Let's assume we understand how to calculate the error function and determine if it changes—that's clear.
- But how do we change the parameters so that the algorithm makes fewer mistakes?
  - Problem. There are a lot of parameters.

Error Function

- Alright. Let's assume we understand how to calculate the error function and determine if it changes—that's clear.
- But how do we change the parameters so that the algorithm makes fewer mistakes?
  - Problem. There are a lot of parameters.
  - Problem. There is also a lot of data.

Error Function

- Alright. Let's assume we understand how to calculate the error function and determine if it changes—that's clear.
- But how do we change the parameters so that the algorithm makes fewer mistakes?
  - Problem. There are a lot of parameters.
  - Problem. There is also a lot of data.
  - Problem. Parameters need to be updated not only correctly but also quickly.

Calculus. Motivation

- And it turns out that the answers to these (and many other) questions lie in a branch of mathematics called calculus; more specifically, in the sub-branch of calculus known as "differential calculus" (of scalar functions of many variables).
- This three-hundred-year-old science was originally created to meet the needs of physicists; nevertheless, it remains relevant to this day, as without it, any machine learning would simply be impossible.

Calculus. Motivation

- That's why in today's lecture, we'll first revisit the key concepts of differential calculus.
- Then, we'll see how these naturally lead to what are known as first-order methods for solving problems of unconstrained continuous optimization.

Calculus. Motivation

- That's why in today's lecture, we'll first revisit the key concepts of differential calculus.
- Then, we'll see how these naturally lead to what are known as first-order methods for solving problems of unconstrained continuous optimization.
- Well, are you ready?
- Let's go! :)

Basic Concepts

Differential calculus

- Differential calculus is a branch of calculus.

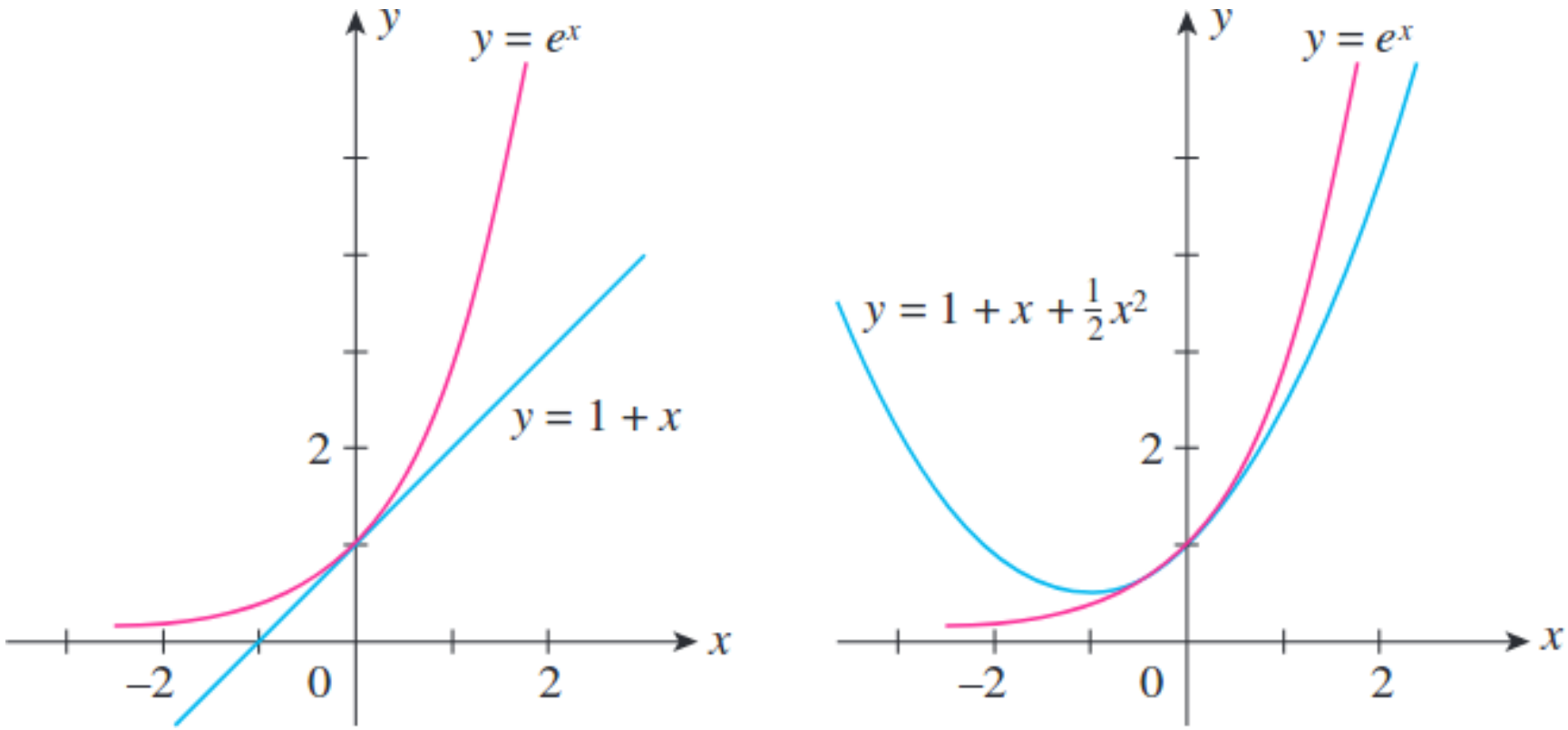
Differential calculus

- Differential calculus is a branch of calculus.
- The original tasks of this science are:
  - Analyzing the behavior of functions in a small neighborhood of a point;

Differential calculus

- Differential calculus is a branch of calculus.
- The original tasks of this science are:
  - Analyzing the behavior of functions in a small neighborhood of a point;
  - Locally approximating complex functions with simple and understandable mathematical objects like polynomials or trigonometric sums.

Differential calculus



First and second order approximations for the function y = e^x in the vicinity of the point 0.

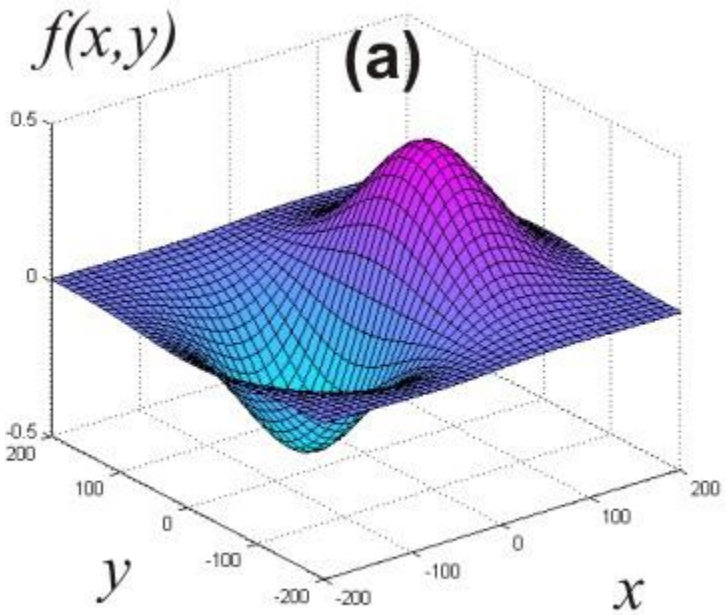
Differential calculus

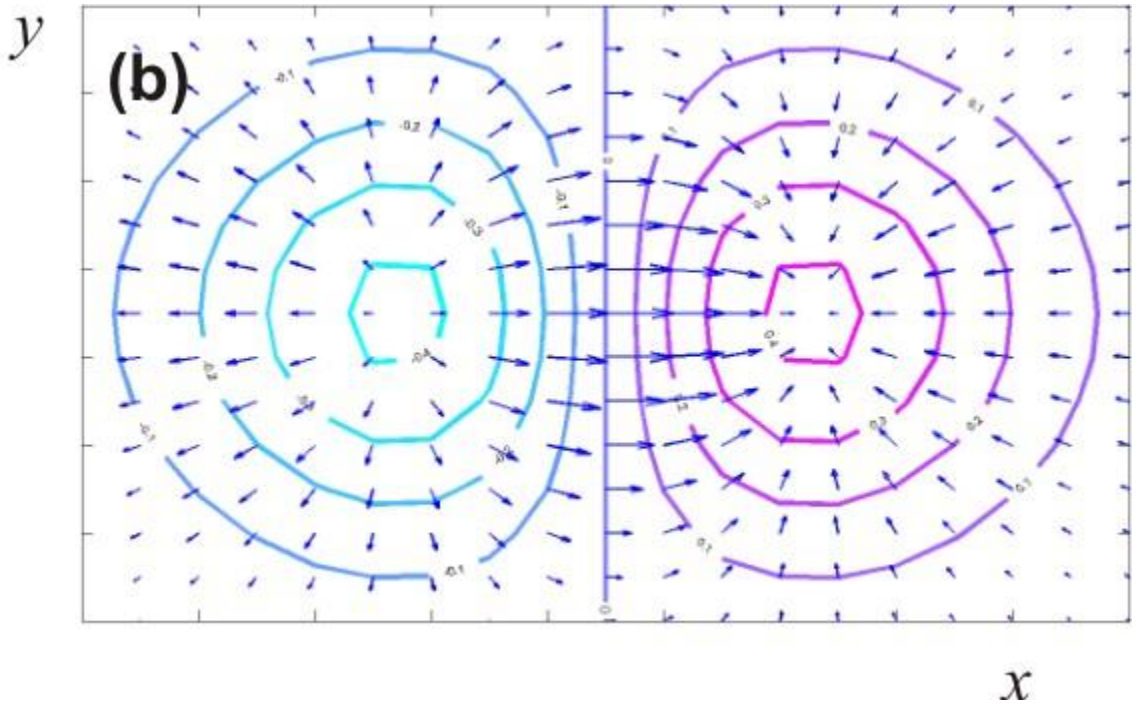
- For most machine learning methods, it is sufficient to understand what the first derivative is.
- Knowing the first derivative at a point allows you to construct a linear approximation of the function in a small neighborhood of that point.

Differential calculus

- For most machine learning methods, it is sufficient to understand what the first derivative is.
- Knowing the first derivative at a point allows you to construct a linear approximation of the function in a small neighborhood of that point.
- It turns out that this is enough to understand in which direction the value of the function increases (or decreases) most rapidly.

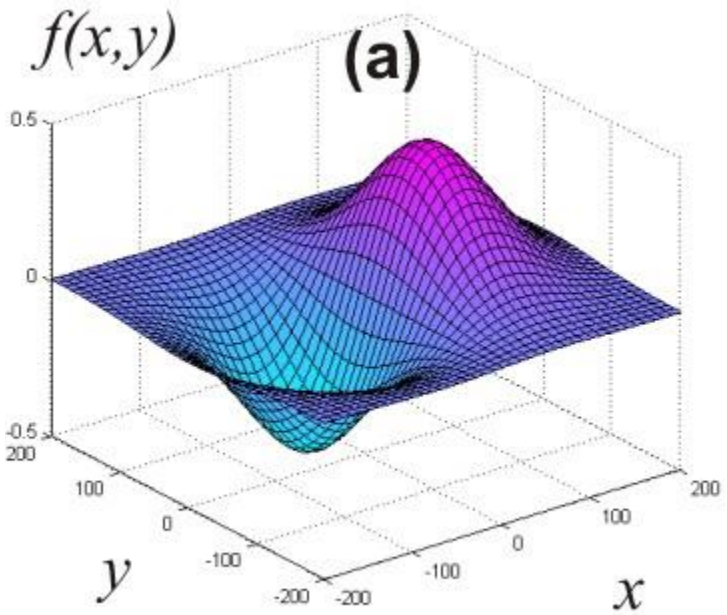
First derivative

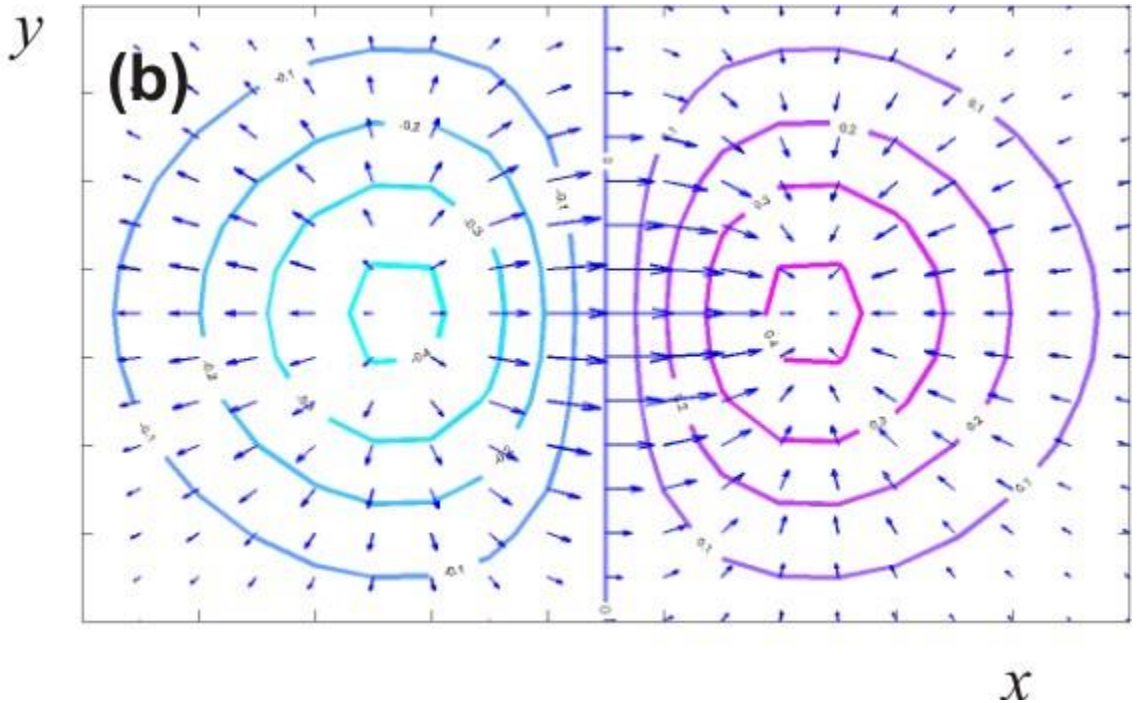




- At the intuitive level, the first derivative allows you to build a "height map" of your function.

First derivative





- At each point, you will know in which direction from it there is the steepest ascent or descent.

First derivative

- So, let's set a goal: to recall and fully review the concept of the first derivative, as well as how it is related to finding the local optimum (minimum or maximum) for sufficiently "smooth" functions.

First derivative

- So, let's set a goal: to recall and fully review the concept of the first derivative, as well as how it is related to finding the local optimum (minimum or maximum) for sufficiently "smooth" functions.
- The concepts of calculus are in strict connection and hierarchy with each other.
- And therefore, to move on to the subject conversation about the derivative, we first need to recall a few more important, preceding concepts.

First derivative

- So, let's set a goal: to recall and fully review the concept of the first derivative, as well as how it is related to finding the local optimum (minimum or maximum) for sufficiently "smooth" functions.
- The concepts of calculus are in strict connection and hierarchy with each other.
- And therefore, to move on to the subject conversation about the derivative, we first need to recall a few more important, preceding concepts.
- Let's do this!

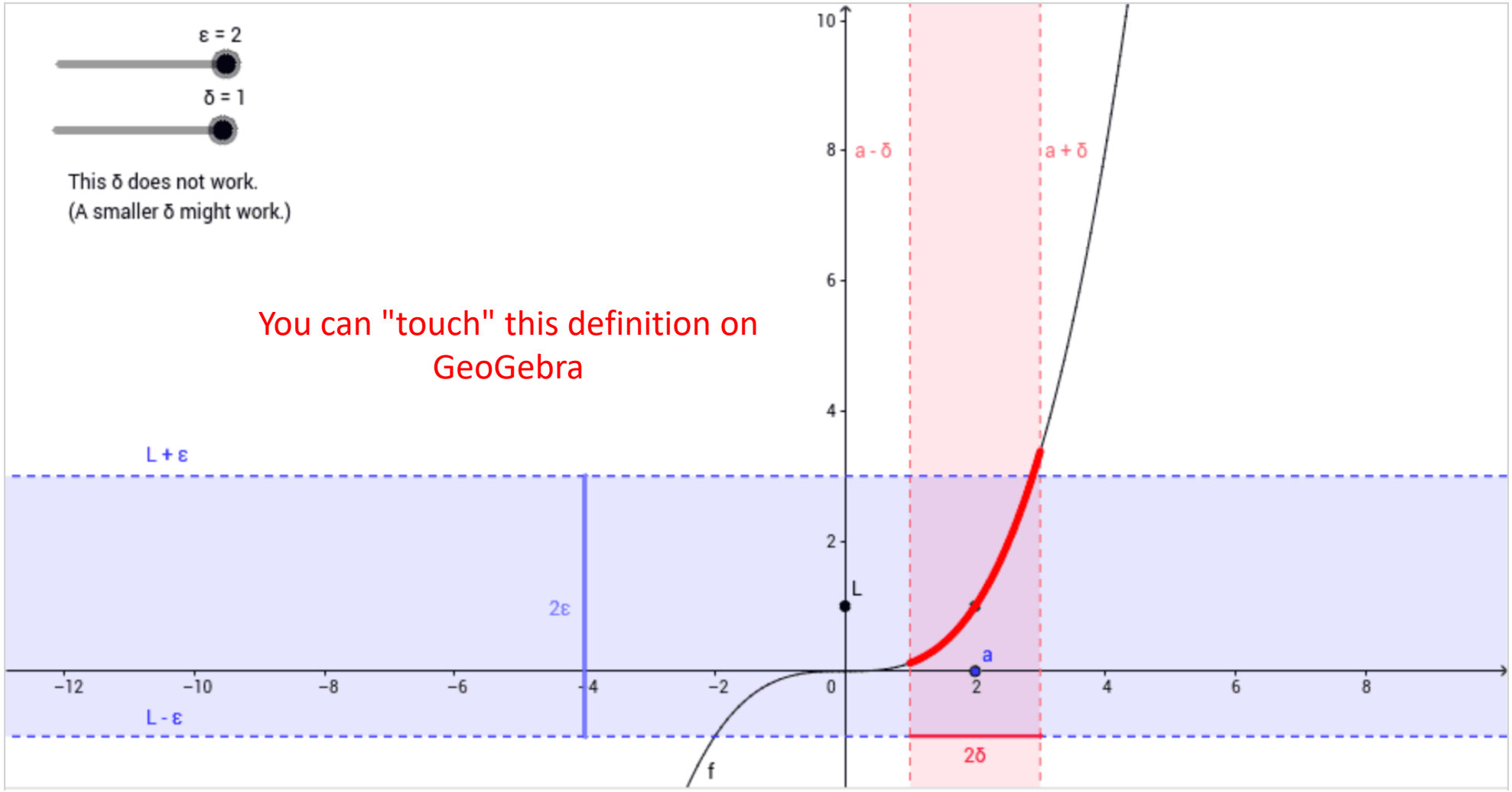
Limit of a function at a point

- Well then, the first such definition will be the definition of the limit of a function at a point.
- Do you remember what that is?

Limit of a function at a point

- Well then, the first such definition will be the definition of the limit of a function at a point.
- Do you remember what that is?
- In your own words:
- If a function f has a limit y_0 at the point x_0, then no matter by what trajectory in the domain of the function you approach the point x_0, the values of the function at the points of the trajectory will get closer and closer to y_0.

Limit of a function at a point



Continuity of a function at a point

- The next concept we'll need is the concept of continuity of a function at a point. This concept is closely related to the concept of a limit (the formal definition is almost the same).
- Who's gonna help us with this?

Continuity of a function at a point

- The next concept we'll need is the concept of continuity of a function at a point. This concept is closely related to the concept of a limit (the formal definition is almost the same).
- Who's gonna help us with this?
- In your own words:
- If the function f is continuous at x_0, then no matter by what trajectory we approach the point x_0, we will obtain in the limit the value $f(x_0)$.

Continuity of a function at a point

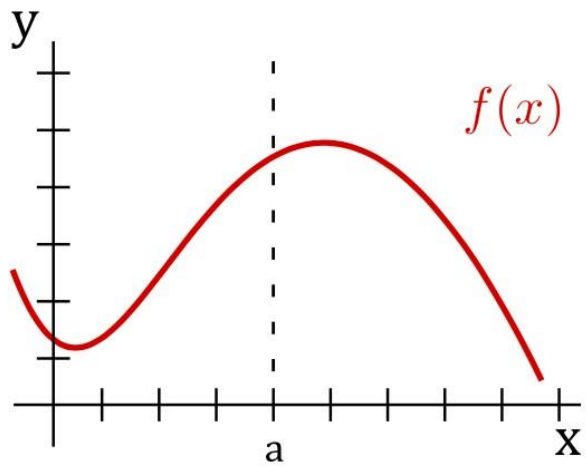
- The next concept we'll need is the concept of continuity of a function at a point. This concept is closely related to the concept of a limit (the formal definition is almost the same).
- Who's gonna help us with this?
- In your own words:
- If the function f is continuous at x_0, then no matter by what trajectory we approach the point x_0, we will obtain in the limit the value $f(x_0)$.
  - That is, essentially, in the definition of a limit, we simply replace y_0 with $f(x_0)$.

Continuity of a function at a point

- Most of the functions we will deal with in this course are continuous.
- However, this is far from being a "free" property.
- It is often violated, but at the same time, it is almost always required.
- Be sure to pay attention to this in your reasoning.

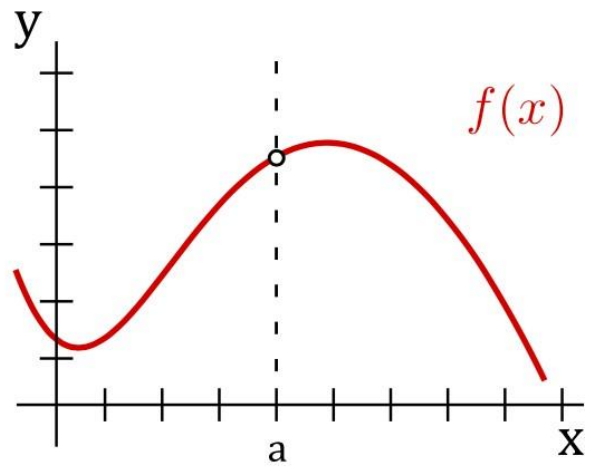
Continuity of a function at a point

- Most of the functions we will deal with in this course are continuous.
- However, this is far from being a "free" property.
- It is often violated, but at the same time, it is almost always required.
- Be sure to pay attention to this in your reasoning.
- By the way, how can we classify the cases when a function is not continuous?



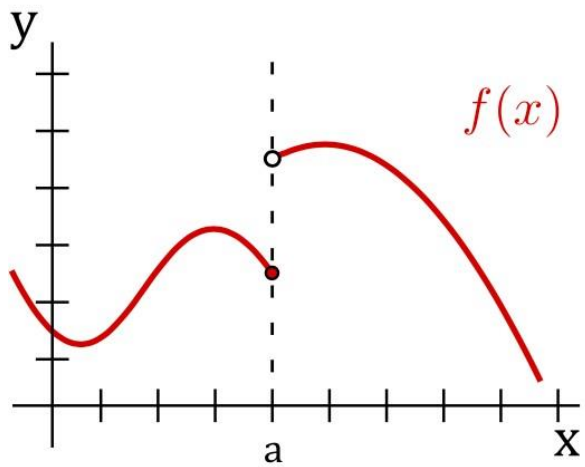
continuous at x = a

$\left(\lim_{x \to a} f(x) = f(a) \right)$



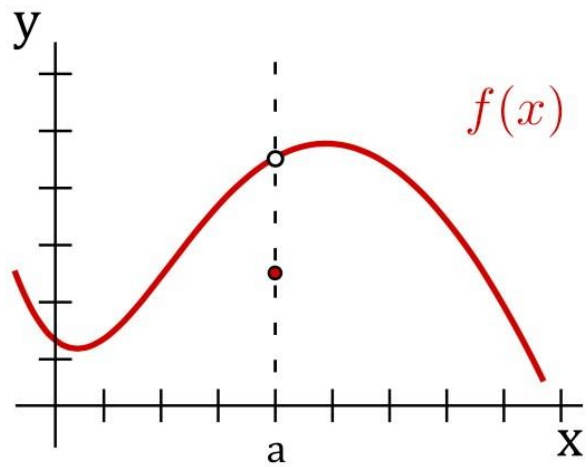
f(a) not defined

(i) fails to hold



$\lim_{x \to a} f(x)$ does not exist

(ii) fails to hold



$\lim_{x \to a} f(x) \neq f(a)$

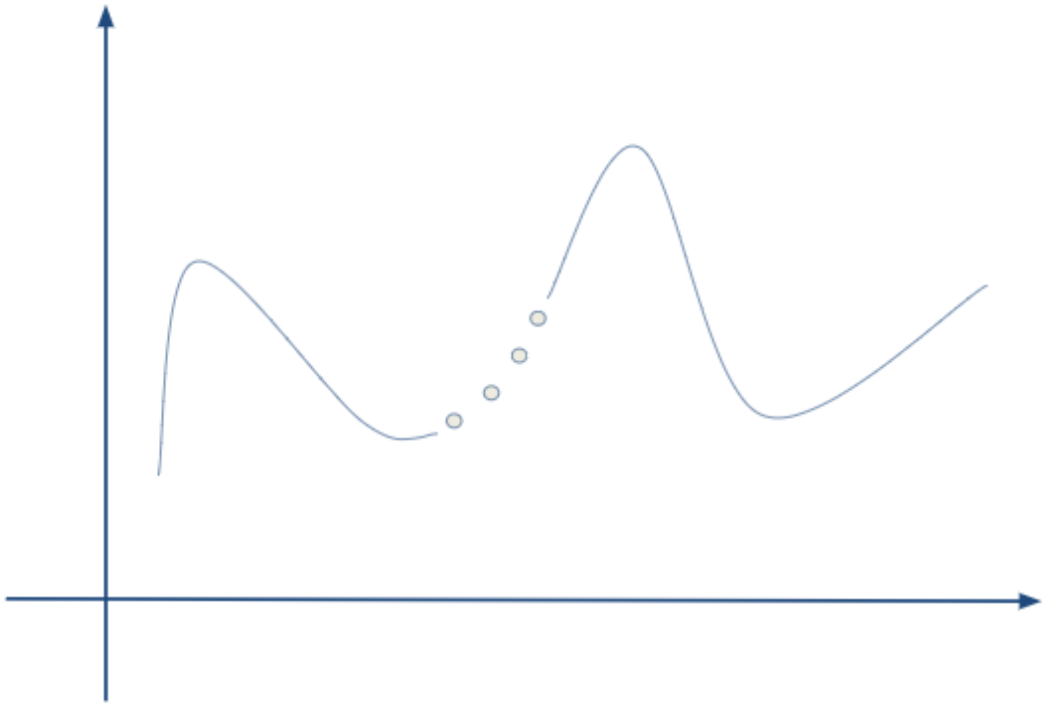
(iii) fails to hold

Continuity of a function on a set

- What about the continuity of a function on a set?

Continuity of a function on a set

- What about the continuity of a function on a set?
- A function is said to be continuous on a set if it is continuous at every point of that set (the set can be a simple segment on the axis, for example).



Differentiability

- What is differentiability?

Differentiability

- What is differentiability?
- If a function is well approximated by a linear dependence on parameters in the vicinity of a point, it is said to be differentiable at that point.
- It is said that a function is differentiable on a set if it is differentiable at every interior point of that set.

Differentiability and continuity

- Continuous functions have a multitude of pleasant properties, which we unfortunately will not have time to discuss in the scope of this course.
- For us, it's important to know that continuity is a necessary condition for differentiability (which is why we're all here): if a function is differentiable on a set, then it is necessarily continuous on it.

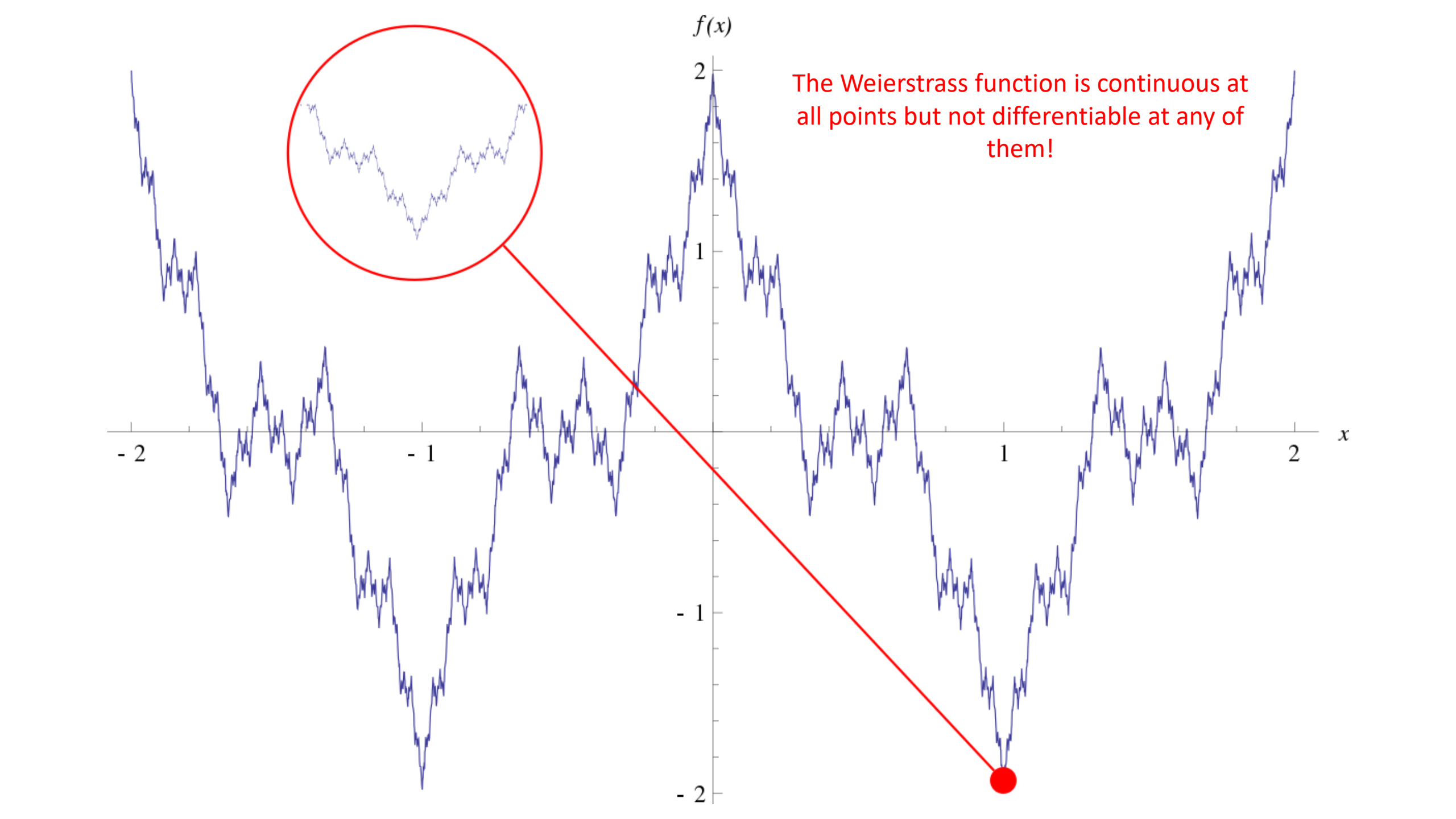
Differentiability and continuity

- Continuous functions have a multitude of pleasant properties, which we unfortunately will not have time to discuss in the scope of this course.
- For us, it's important to know that continuity is a necessary condition for differentiability (which is why we're all here): if a function is differentiable on a set, then it is necessarily continuous on it.
- The converse is generally not true.

Differentiability and continuity

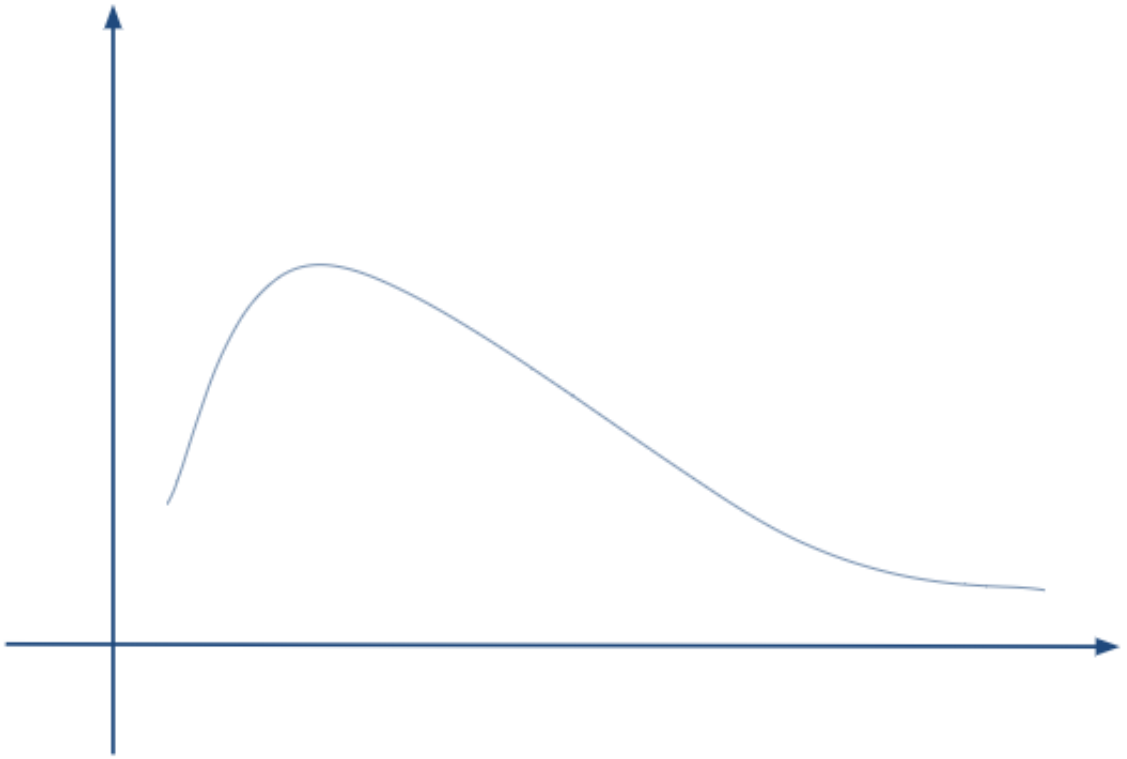
- Continuous functions have a multitude of pleasant properties, which we unfortunately will not have time to discuss in the scope of this course.
- For us, it's important to know that continuity is a necessary condition for differentiability (which is why we're all here): if a function is differentiable on a set, then it is necessarily continuous on it.
- The converse is generally not true.

Maybe you can even provide a counterexample?


f(x)
2
1
- 1
- 2
- 2
- 1
1
2
x
The Weierstrass function is continuous at all points but not differentiable at any of them!

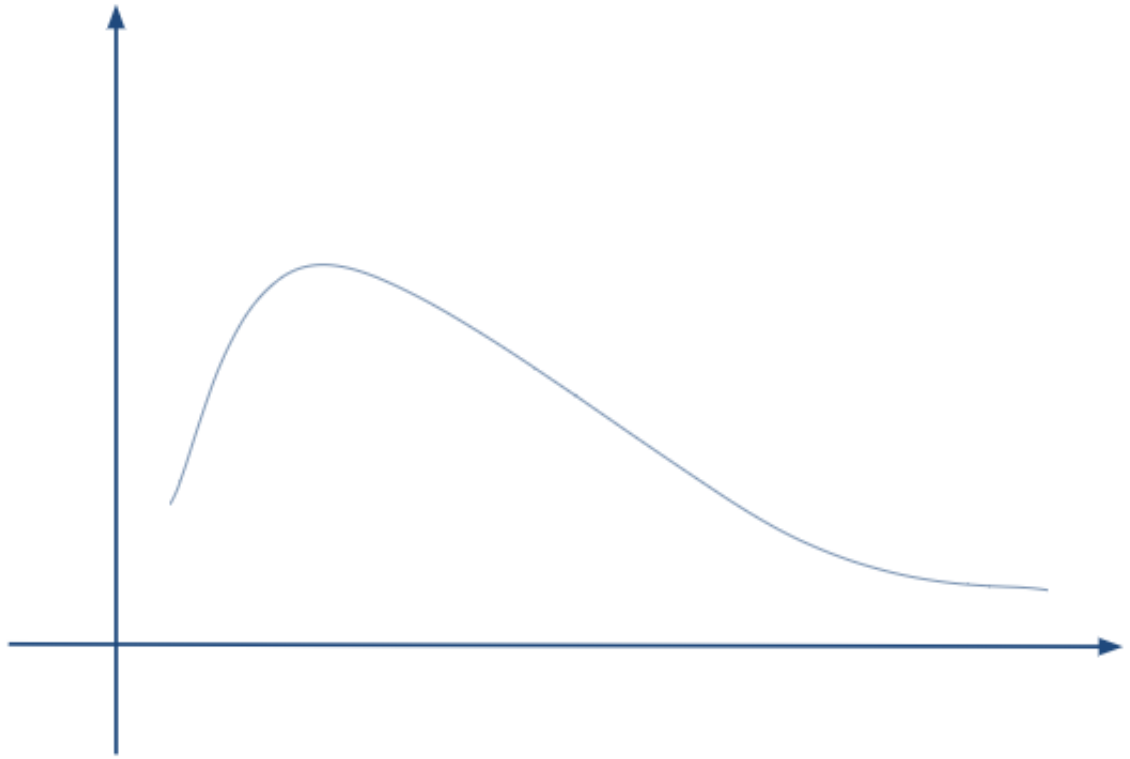
First derivative

- But let's return to the derivative!



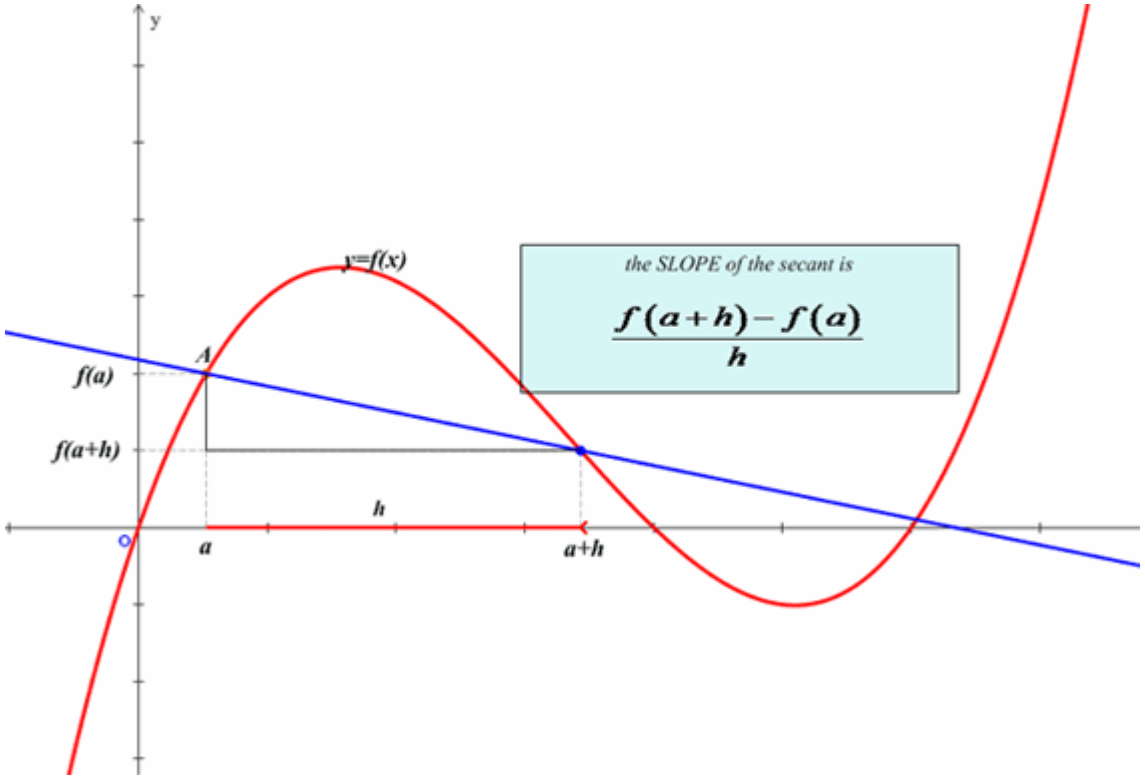
First derivative

- But let's return to the derivative!
- It's convenient to think of the derivative as the instantaneous speed of a material point whose trajectory is defined by the function $f(x)$



First derivative

- The derivative at a point uniquely defines the tangent to the function near that point and is numerically equal to the tangent of the angle of its slope.



First derivative

- There is a set of formal rules for calculating derivatives. Moreover, there are automatic differentiation packages like sympy, torch.autograd, and jax, which will do everything for you.

First derivative

- There is a set of formal rules for calculating derivatives. Moreover, there are automatic differentiation packages like sympy, torch.autograd, and jax, which will do everything for you.
- However, it is extremely important to understand that before substituting values of the arguments into the derivative formula, it is necessary to ensure that the function is differentiable at the point in question!

First derivative

- From the relationship of the derivative with the tangent line equation at a point, it clearly follows that:
- If a function is differentiable over an interval, one can find intervals of its increase, decrease, and constancy.

First derivative

- From the relationship of the derivative with the tangent line equation at a point, it clearly follows that:
- If a function is differentiable over an interval, one can find intervals of its increase, decrease, and constancy.
  - If the tangent of the angle of the tangent line is positive, then the function is increasing.
  - If it is negative, then the function is decreasing.
  - If it is zero, then the function is constant in the vicinity of the point.

First derivative

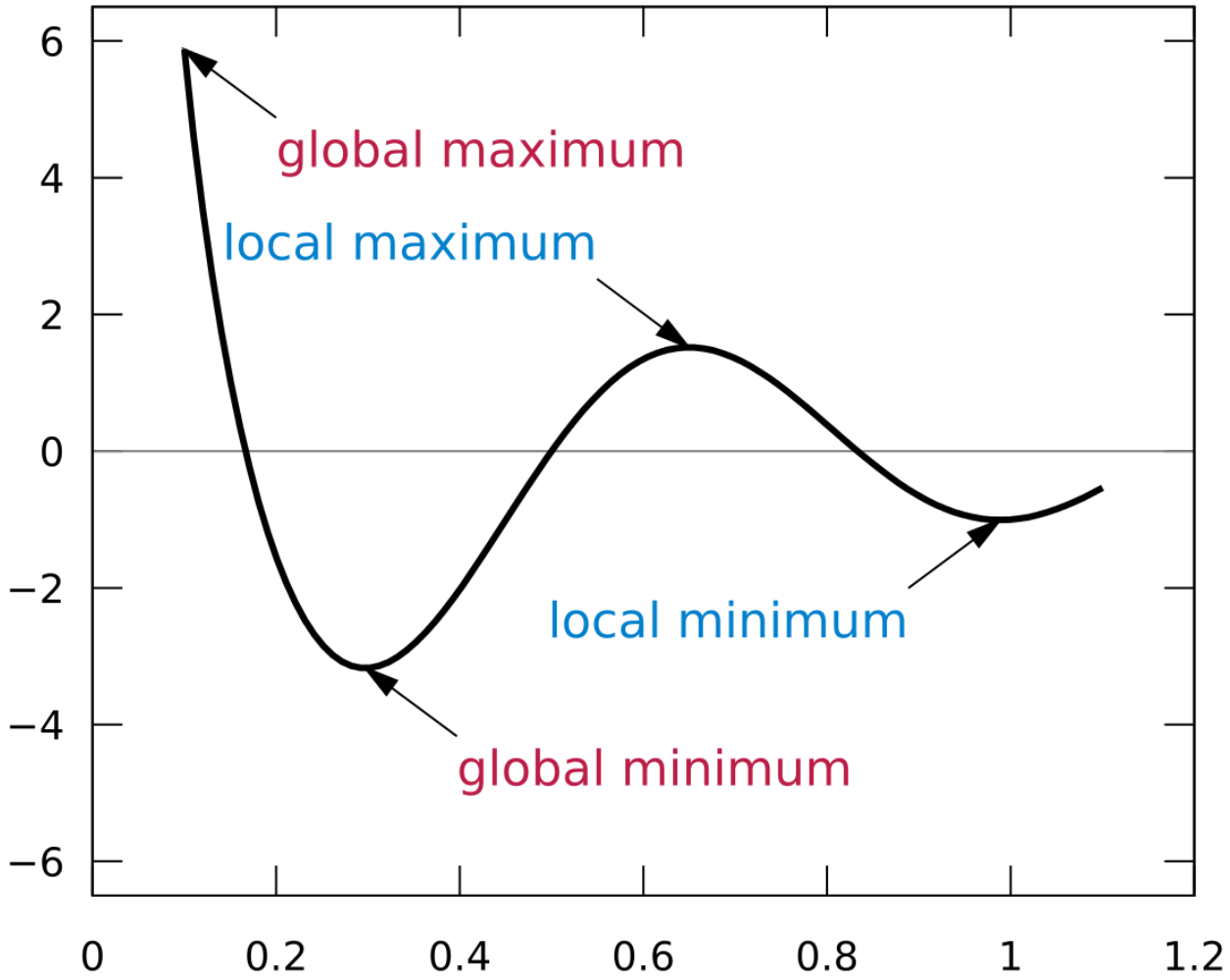
- This wonderful fact allows us to determine a necessary condition for the presence of a local extremum in terms of the first derivative!

First derivative

- This wonderful fact allows us to determine a necessary condition for the presence of a local extremum in terms of the first derivative!

Let's say for the function $f: X \to \mathbb{R}$, the point x is a point of local extremum, then if the function is continuously differentiable in the vicinity of this point, its derivative at this point equals zero: $df(x) = 0$

Extremum points in the one-dimensional case



Extremum points in the one-dimensional case

- Accordingly, the first step in the analytical search for local optima of a differentiable function involves finding the roots of the function's derivative—these are called stationary points.

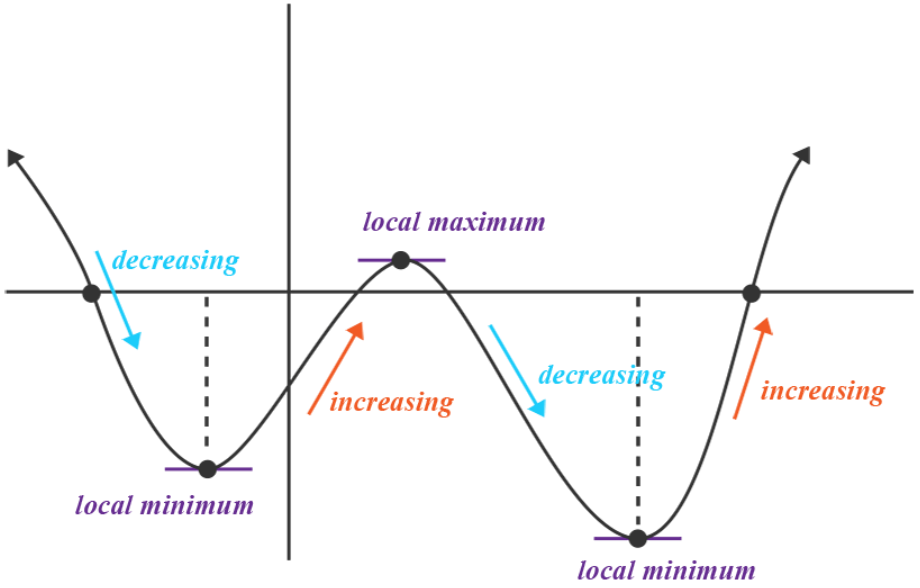
Extremum points in the one-dimensional case

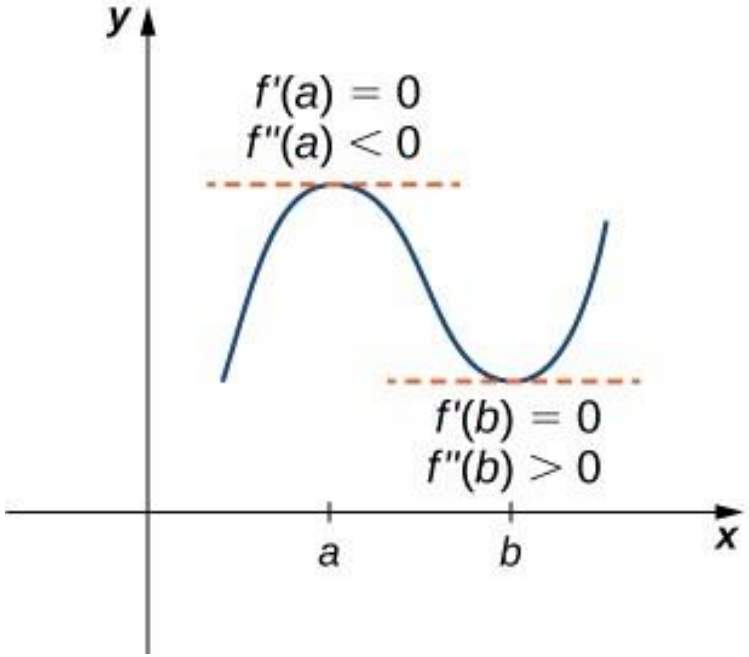
- Accordingly, the first step in the analytical search for local optima of a differentiable function involves finding the roots of the function's derivative—these are called stationary points.

- Note: The second and subsequent derivatives are defined recursively: the (n+1)-th derivative of a function is the derivative of its n-th derivative.

Extremum points in the one-dimensional case

- If a function is concave upwards near a stationary point, it has a local maximum at that point.
- If it is concave downwards, then it has a local minimum. Otherwise, there is no optimum!





Convexity

- At the same time, convexity is an important concept in its own right.
- Let's discuss it separately.

Convexity

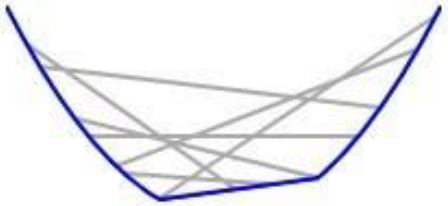
- At the same time, convexity is an important concept in its own right.
- Let's discuss it separately.
- Definition?

Convexity

- At the same time, convexity is an important concept in its own right.
- Let's discuss it separately.
- Definition?
- Consider a function f and a secant line passing through the points $\big(x_1, f(x_1)\big), \big(x_2, f(x_2)\big), x_1 < x_2$.
- If the graph of f on the interval (x_1, x_2) lies strictly below the secant, then the function f is concave downwards on this interval, and if it lies strictly above the secant, it is concave upwards.

Convexity

A concave function:
no line segment joining
two points on the graph
lies above the graph
at any point



A convex function:
no line segment joining
two points on the graph
lies below the graph
at any point

A function that is neither
concave nor convex:
the line segment shown lies
above the graph at some
points and below it at others

- Consider a function f and a secant line passing through the points $\big(x_1, f(x_1)\big), \big(x_2, f(x_2)\big), x_1 < x_2$.
- If the graph of f on the interval (x_1, x_2) lies strictly below the secant, then the function f is concave downwards on this interval, and if it lies strictly above the secant, it is concave upwards.

Differential calculus

- Okay, we have successfully reviewed the main concepts of calculus and even recalled something about derivatives and mathematical optimization, which is certainly pleasing!
- But what is the main problem—or, one might say, the main understatement—in the context of the picture we are currently looking at in differential calculus?

Differential calculus

- Okay, we have successfully reviewed the main concepts of calculus and even recalled something about derivatives and mathematical optimization, which is certainly pleasing!
- But what is the main problem—or, one might say, the main understatement—in the context of the picture we are currently looking at in differential calculus?
- Let's give a brief answer here—

Differential calculus

- Okay, we have successfully reviewed the main concepts of calculus and even recalled something about derivatives and mathematical optimization, which is certainly pleasing!
- But what is the main problem—or, one might say, the main understatement—in the context of the picture we are currently looking at in differential calculus?
- Let's give a brief answer here—unidimensionality.

Multidimensional optimization

Multidimensional optimization

- Despite our previous discussions about single-variable functions, in reality, you will almost never have to deal with such functions.
- The task we want to solve—to find the optimal parameters of a machine learning algorithm—requires us to determine the derivative of a function of many variables.

Multidimensional optimization

- Let's note right away that our intuitive understanding of the derivative as the instantaneous speed fails if the function depends on at least two arguments.

Multidimensional optimization

- Let's note right away that our intuitive understanding of the derivative as the instantaneous speed fails if the function depends on at least two arguments.
- Let's give an example. Imagine you are standing on a hill and want to slide down it on a sled.

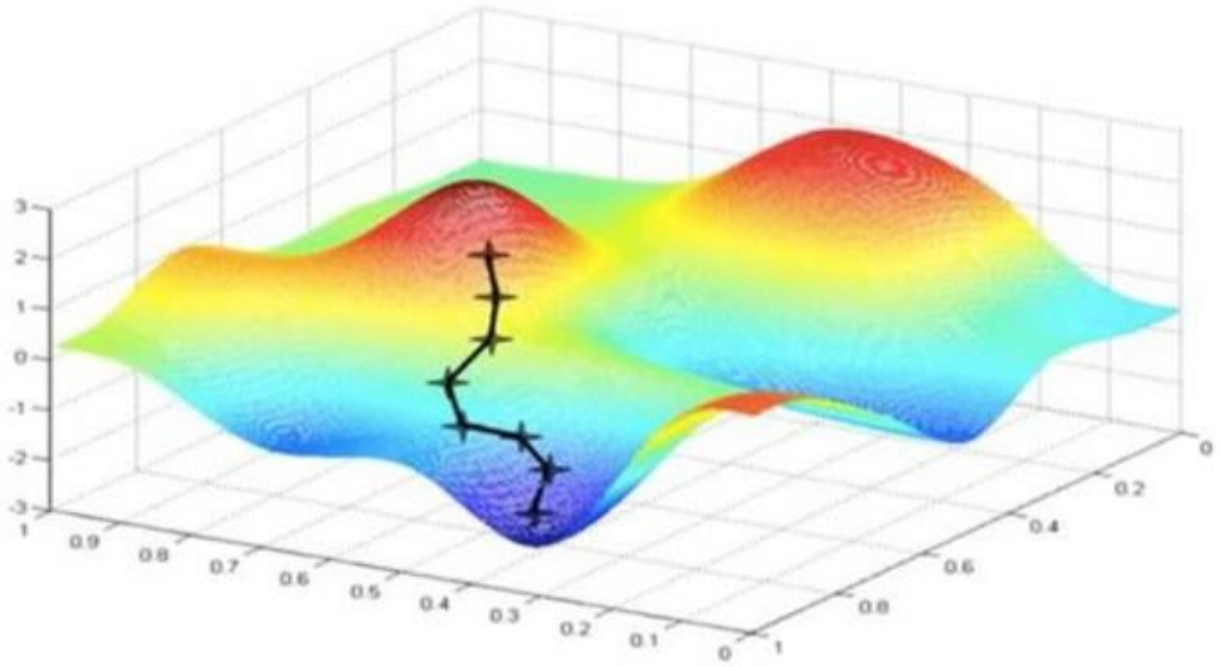
Multidimensional optimization

- Let's note right away that our intuitive understanding of the derivative as the instantaneous speed fails if the function depends on at least two arguments.
- Let's give an example. Imagine you are standing on a hill and want to slide down it on a sled.
- Look around: the speed of your movement will depend on the chosen direction.

Multidimensional optimization

- Let's note right away that our intuitive understanding of the derivative as the instantaneous speed fails if the function depends on at least two arguments.
- Let's give an example. Imagine you are standing on a hill and want to slide down it on a sled.
- Look around: the speed of your movement will depend on the chosen direction.
- Moreover, you can slide down to the same point not just in two ways, as before (from the left and from the right), but along countless trajectories!

Multidimensional optimization



Now the instantaneous speed depends not only on the point,
but also on the direction of motion.

Directional derivative

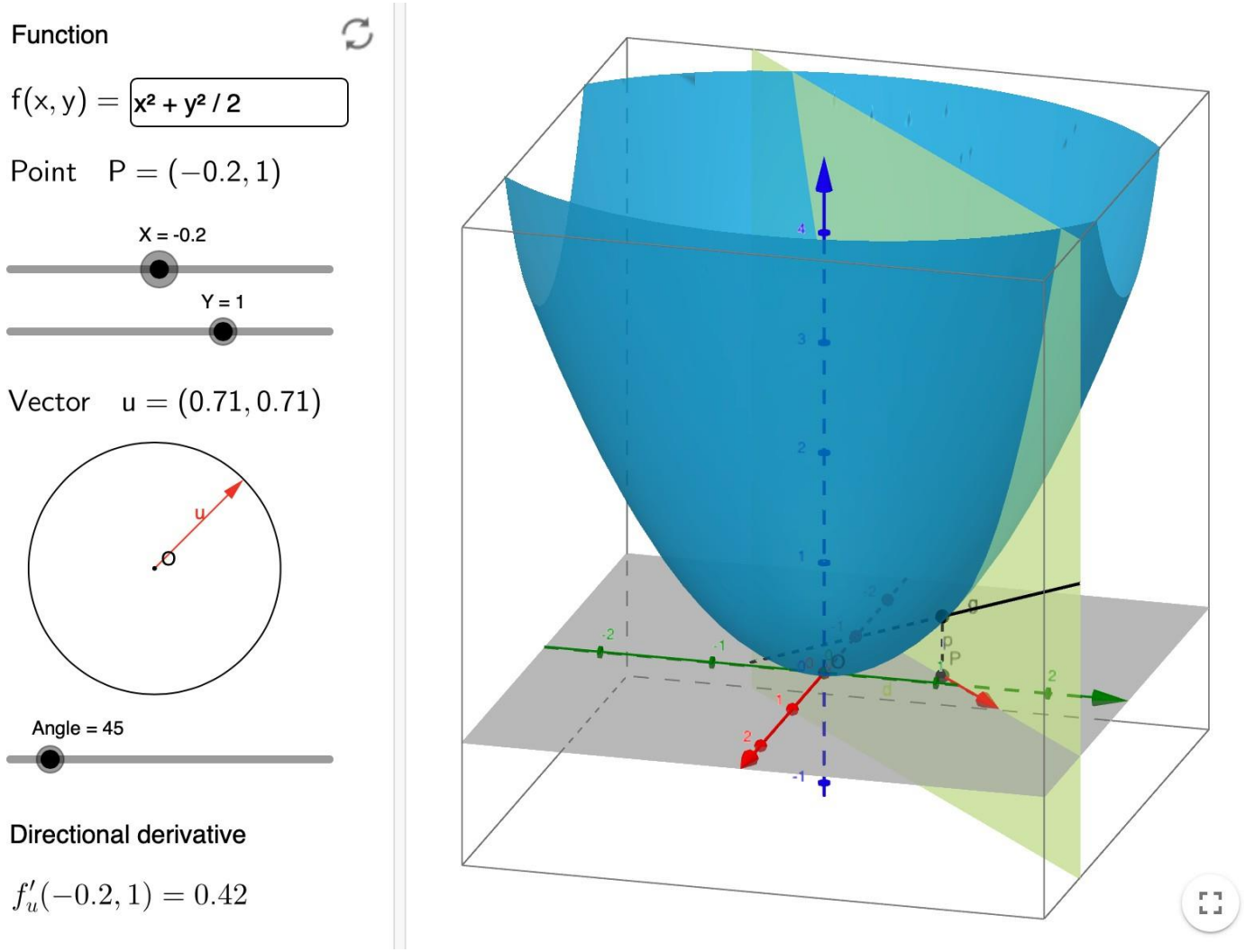
- This observation naturally motivates the concept of the directional derivative.
- What is it?

Directional derivative

- This observation naturally motivates the concept of the directional derivative.
- What is it?
- You can take the projection of a multivariable function in a particular direction and obtain a familiar function of one variable! The result of taking the derivative of such a function will be the directional derivative.

Directional derivative

You can "touch" this definition on GeoGebra



Partial derivative

- What is a partial derivative?

Partial derivative

- What is a partial derivative?
- The directional derivatives along each of the main axes are called partial derivatives.

Partial derivative

- What is a partial derivative?
- The directional derivatives along each of the main axes are called partial derivatives.
- Partial derivatives are significantly easier to calculate because the values of the other variables can be considered fixed, and consequently, "ignored" during the computation.

Partial derivative

- What is a partial derivative?
- The directional derivatives along each of the main axes are called partial derivatives.
- Partial derivatives are significantly easier to calculate because the values of the other variables can be considered fixed, and consequently, "ignored" during the computation.
- Let's practice a bit and find the partial derivatives, for example, for the function:

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

Gradient

- So now, let's address the most crucial concept for the mathematics of machine learning — the gradient! What is a gradient?

Gradient

- So now, let's address the most crucial concept for the mathematics of machine learning — the gradient! What is a gradient?
- A covector composed of partial derivatives is called a gradient.

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$

Gradient

- So now, let's address the most crucial concept for the mathematics of machine learning — the gradient! What is a gradient?
- A covector composed of partial derivatives is called a gradient.

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$

- How are directional derivatives related to the gradient?

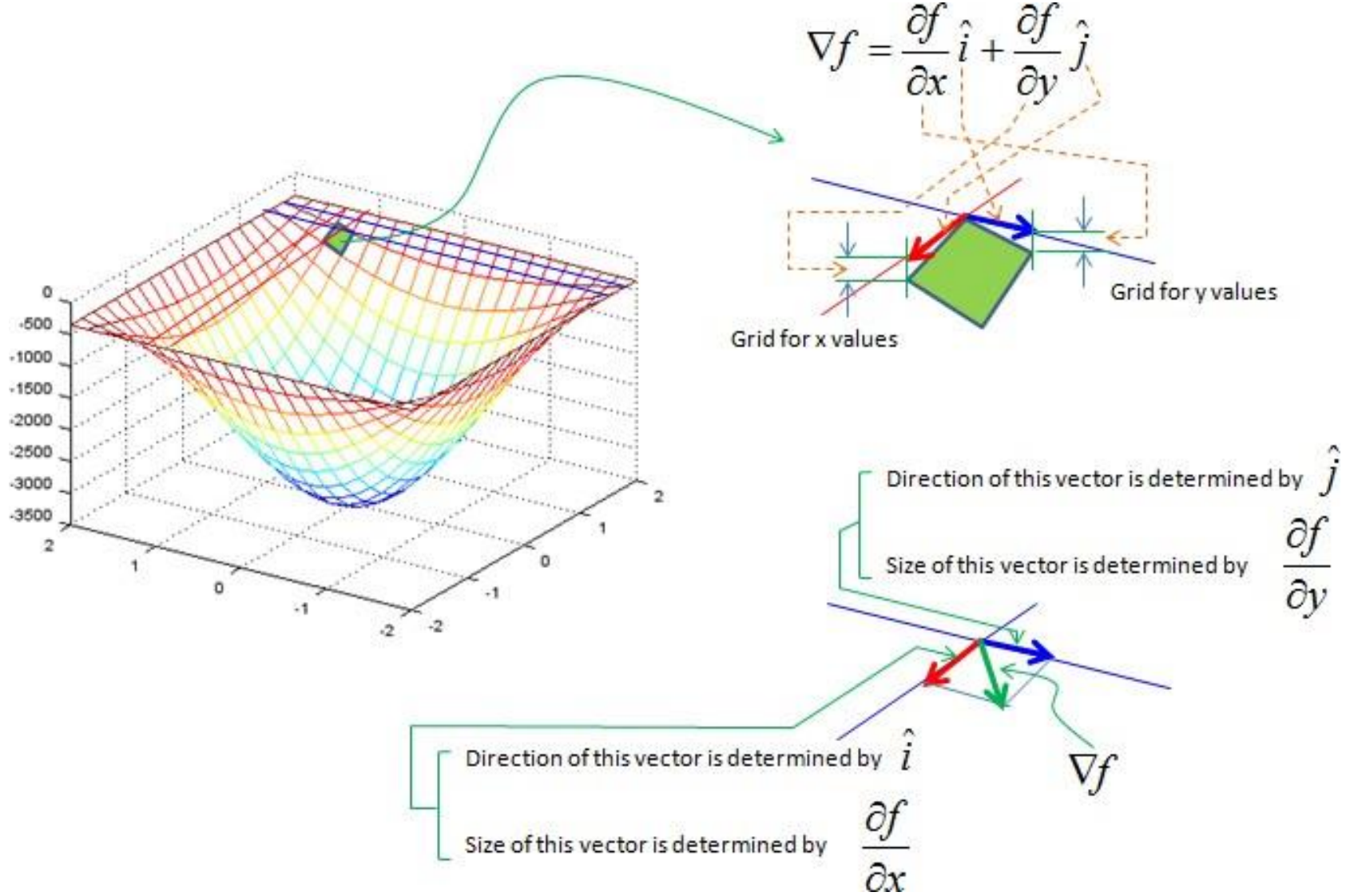
Gradient

- So now, let's address the most crucial concept for the mathematics of machine learning — the gradient! What is a gradient?
- A covector composed of partial derivatives is called a gradient.

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)$$

- How are directional derivatives related to the gradient?
- The derivative in any direction can be calculated through the dot product of the gradient with the directional vector of that direction.

Gradient


$\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j}$
Grid for y values
Grid for x values
Direction of this vector is determined by $\hat{j}$
Size of this vector is determined by $\frac{\partial f}{\partial y}$
Direction of this vector is determined by $\hat{i}$
Size of this vector is determined by $\frac{\partial f}{\partial x}$
∇f

Multidimensional optimization

- We will skip discussions on the conditions under which a function of many variables is differentiable, and instead focus on what we obtain if the function is differentiable.

Multidimensional optimization

- We will skip discussions on the conditions under which a function of many variables is differentiable, and instead focus on what we obtain if the function is differentiable.
- If you know that a function is differentiable at a point, the direction of the gradient at this point gives you very important information!
  - And for most standard functions in ML, you know this!

Multidimensional optimization

- We will skip discussions on the conditions under which a function of many variables is differentiable, and instead focus on what we obtain if the function is differentiable.
- If you know that a function is differentiable at a point, the direction of the gradient at this point gives you very important information!
  - And for most standard functions in ML, you know this!
- What information is that?

Multidimensional optimization

The gradient of a scalar field (a scalar function of many variables) is oriented along the direction of the fastest increase of the function in the vicinity of that point!

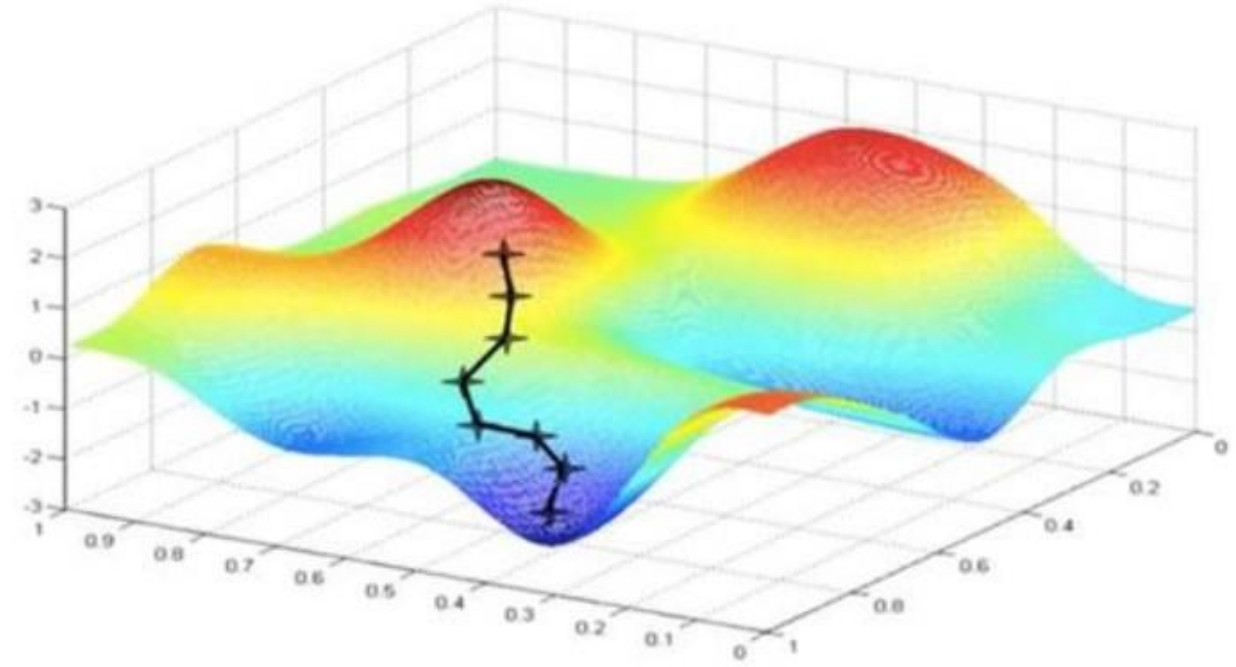
Multidimensional optimization

The gradient of a scalar field (a scalar function of many variables) is oriented along the direction of the fastest increase of the function in the vicinity of that point!

- Hooray! If the function is differentiable, then at each point we know the direction of its fastest increase (or decrease—simply multiply the gradient by -1 to get the negative gradient).

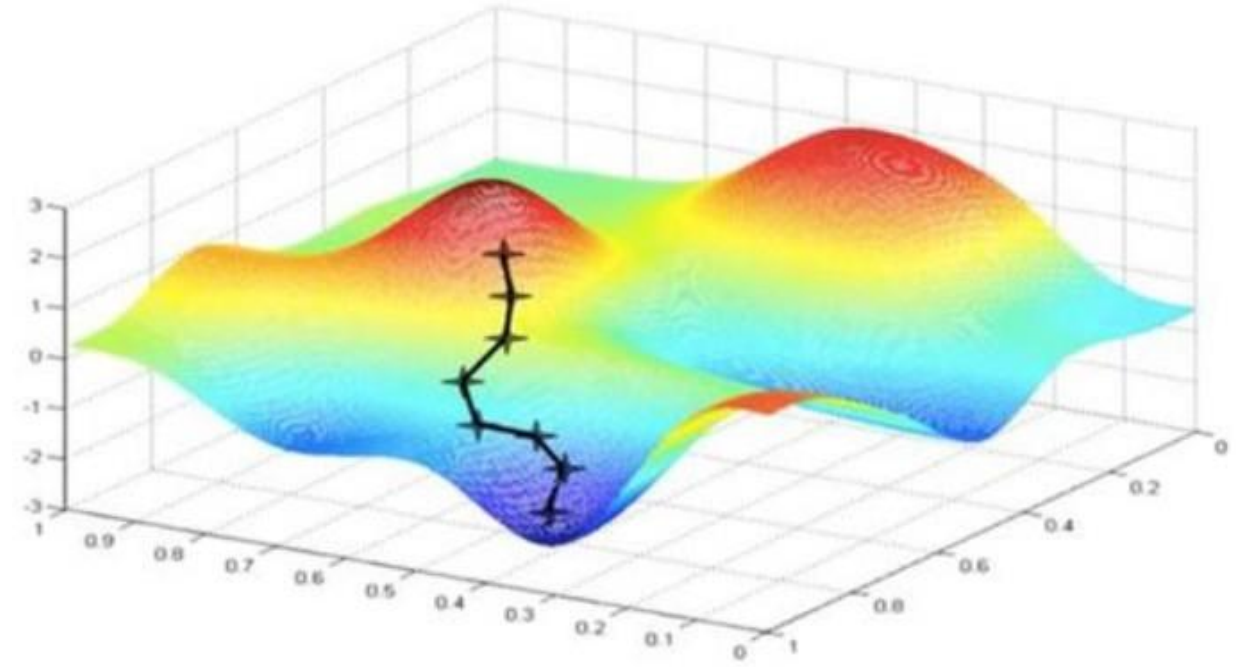
Multidimensional optimization

- Recall our example with the slide and the sled.



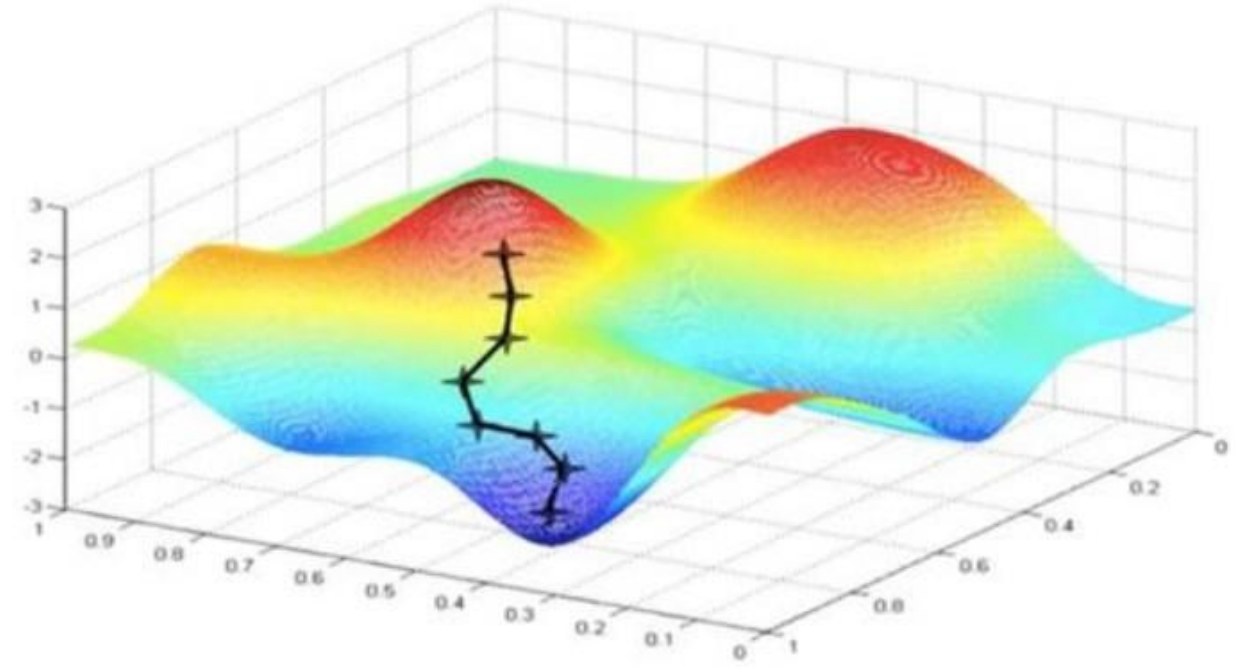
Multidimensional optimization

- Recall our example with the slide and the sled.
- You intuitively know that you'll go downhill fastest if you ride the steepest slopes.



Multidimensional optimization

- To put it in math terms, you need to move along the anti-gradient all the time.



Multidimensional optimization

- Congratulations! We have just invented the method of gradient descent! :)
- Its modification—stochastic gradient descent—is one of the main pillars of neural network ML.

Multidimensional optimization

- Congratulations! We have just invented the method of gradient descent! :)
- Its modification—stochastic gradient descent—is one of the main pillars of neural network ML.
- Of course, this is far from the only optimization algorithm you should know.
- Nevertheless, the most popular algorithms—so-called first-order methods—are modifications of standard gradient descent.

Gradient descent

Gradient descent

- In essence, we have already derived the gradient descent.
- It remains to write it down in formulas.

Gradient descent

- In essence, we have already derived the gradient descent.
- It remains to write it down in formulas.

$$w^{(n+1)} \leftarrow w^{(n)} - \alpha(n) \frac{\nabla \mathcal{L}(w^{(n)}; X, Y)}{\left\| \nabla \mathcal{L}(w^{(n)}; X, Y) \right\|}$$

Gradient descent

- In essence, we have already derived the gradient descent.
- It remains to write it down in formulas.

$$w^{(n+1)} \leftarrow w^{(n)} - \alpha(n) \frac{\nabla \mathcal{L}(w^{(n)}; X, Y)}{\left\| \nabla \mathcal{L}(w^{(n)}; X, Y) \right\|}$$

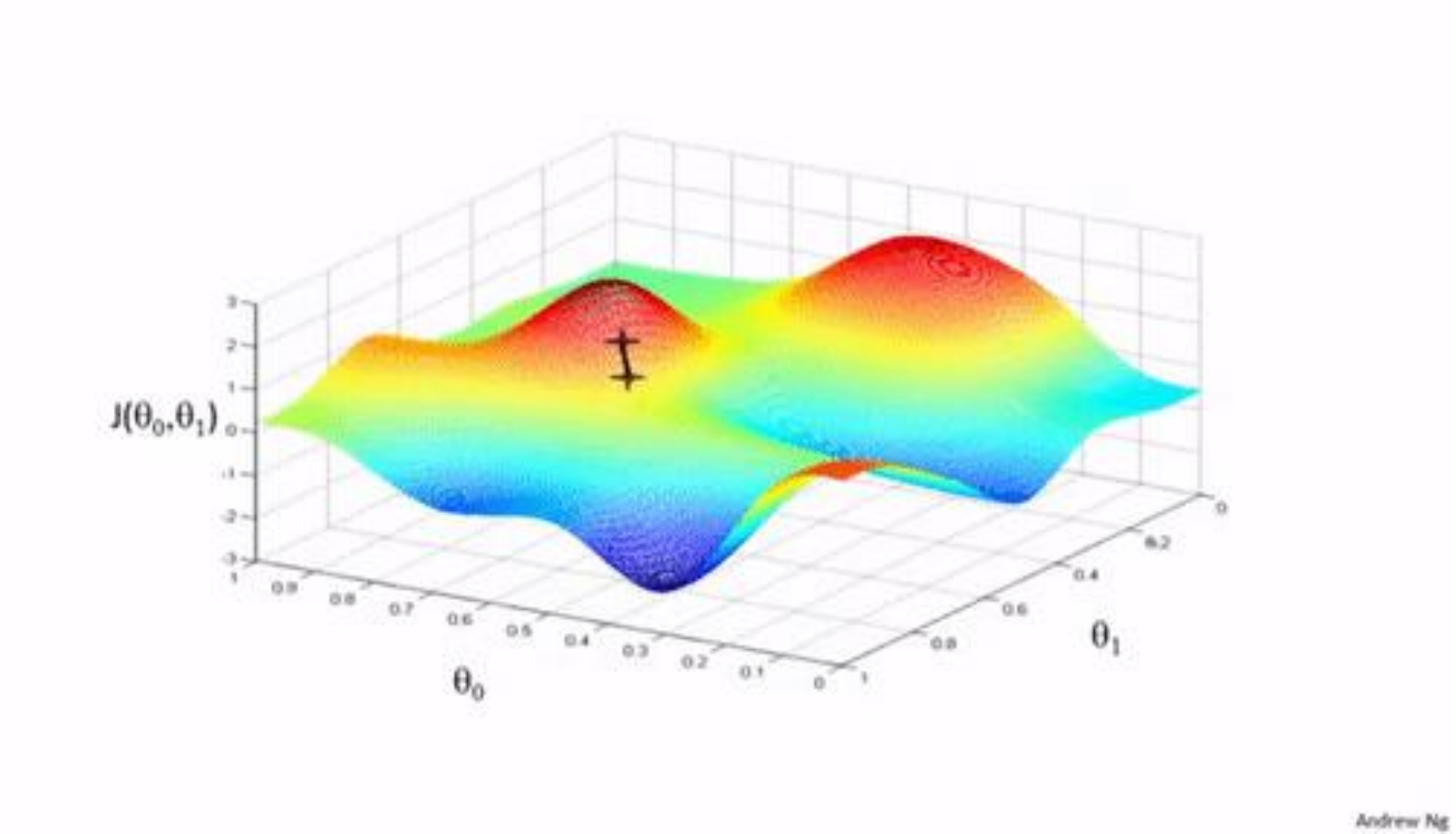
- Scared? :)

Gradient descent

$$w^{(n+1)} \leftarrow w^{(n)} - \alpha(n) \frac{\nabla \mathcal{L}(w^{(n)}; X, Y)}{\left\| \nabla \mathcal{L}(w^{(n)}; X, Y) \right\|}$$

- where
  - $w^{(n)}$ — algorithm parameters at step n;
  - $\frac{\nabla \mathcal{L}(w^{(n)};X,Y)}{\left\| \nabla \mathcal{L}(w^{(n)};X,Y) \right\|}$ — the direction of motion given by the gradient of the error function at the current value of the model parameters, on the training sample X with reference responses Y (if known);
  - $\alpha(n)$ — learning rate, gradient step length.

Gradient descent



Visualization of gradient descent for a function
of two real variables

Gradient descent

- Gradient descent has some fairly obvious drawbacks.
- What do you think they are?

Gradient descent

- Gradient descent has some fairly obvious drawbacks.
- What do you think they are?
    1. At each step, it is necessary to calculate the error function for the entire training dataset (which can be very large, with millions of objects);

Gradient descent

- Gradient descent has some fairly obvious drawbacks.
- What do you think they are?
  1. At each step, it is necessary to calculate the error function for the entire training dataset (which can be very large, with millions of objects);
  2. It's necessary to choose a step length that allows for progress without overshooting local optima;

Gradient descent

- Gradient descent has some fairly obvious drawbacks.
- What do you think they are?
  1. At each step, it is necessary to calculate the error function for the entire training dataset (which can be very large, with millions of objects);
  2. It's necessary to choose a step length that allows for progress without overshooting local optima;
  3. There are no guarantees of finding the global minimum.

Gradient descent

- Gradient descent has some fairly obvious drawbacks.
- What do you think they are?
  1. At each step, it is necessary to calculate the error function for the entire training dataset (which can be very large, with millions of objects);
  2. It's necessary to choose a step length that allows for progress without overshooting local optima;
  3. There are no guarantees of finding the global minimum.

- Drawbacks 1 and 2 are solvable, 3—in general, no, but in many cases important to us—very much so!

Gradient descent

- Just now, discussing the problems, we figured out that it's very desirable not to calculate the full gradient of the error function at every step. But how is this possible?

Gradient descent

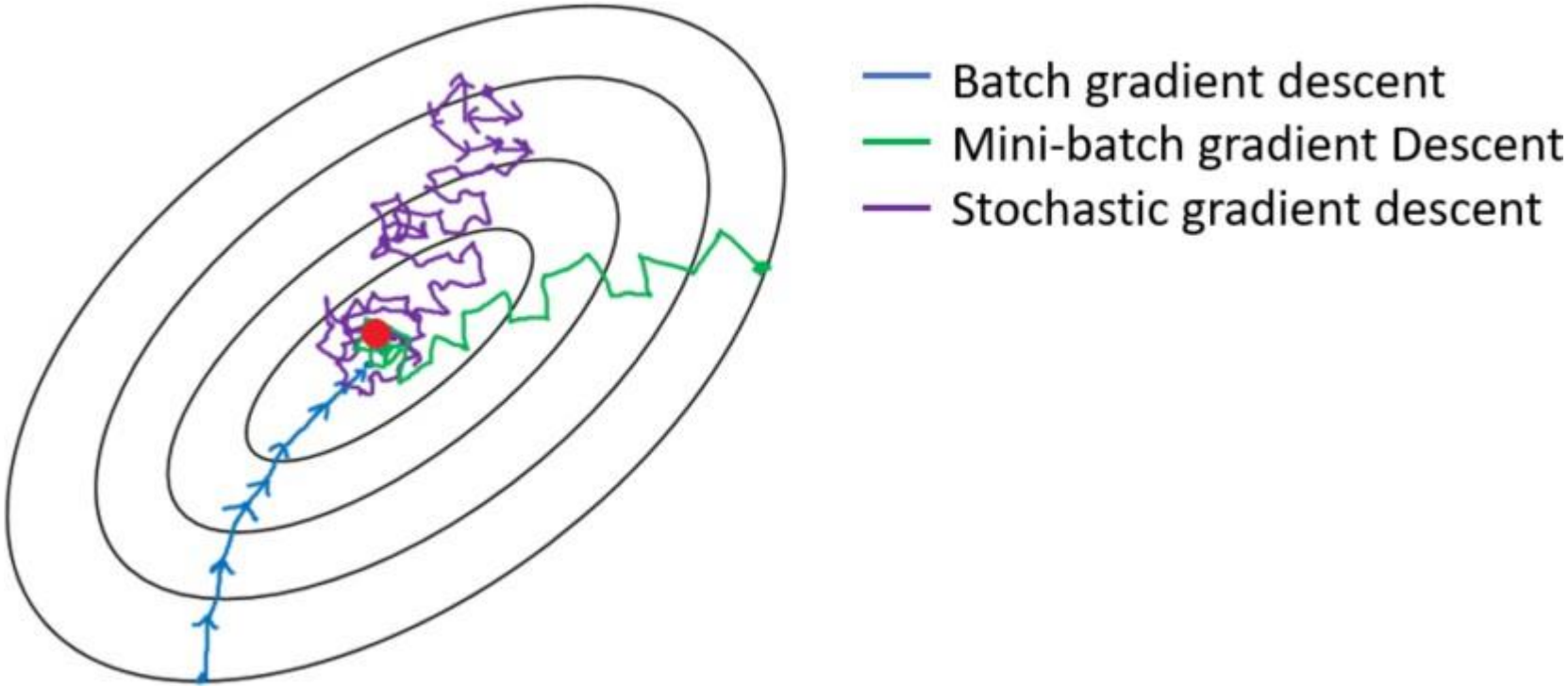
- Just now, discussing the problems, we figured out that it's very desirable not to calculate the full gradient of the error function at every step. But how is this possible?
- One can calculate the gradient on a random subset of objects—referred to as a minibatch.
- This algorithm is known as MiniBatch Gradient Descent.
- If the size of the minibatch is chosen wisely, this modification can converge much faster than the original gradient descent, which is actually very pleasing!

Gradient descent

- There's also stochastic gradient descent....

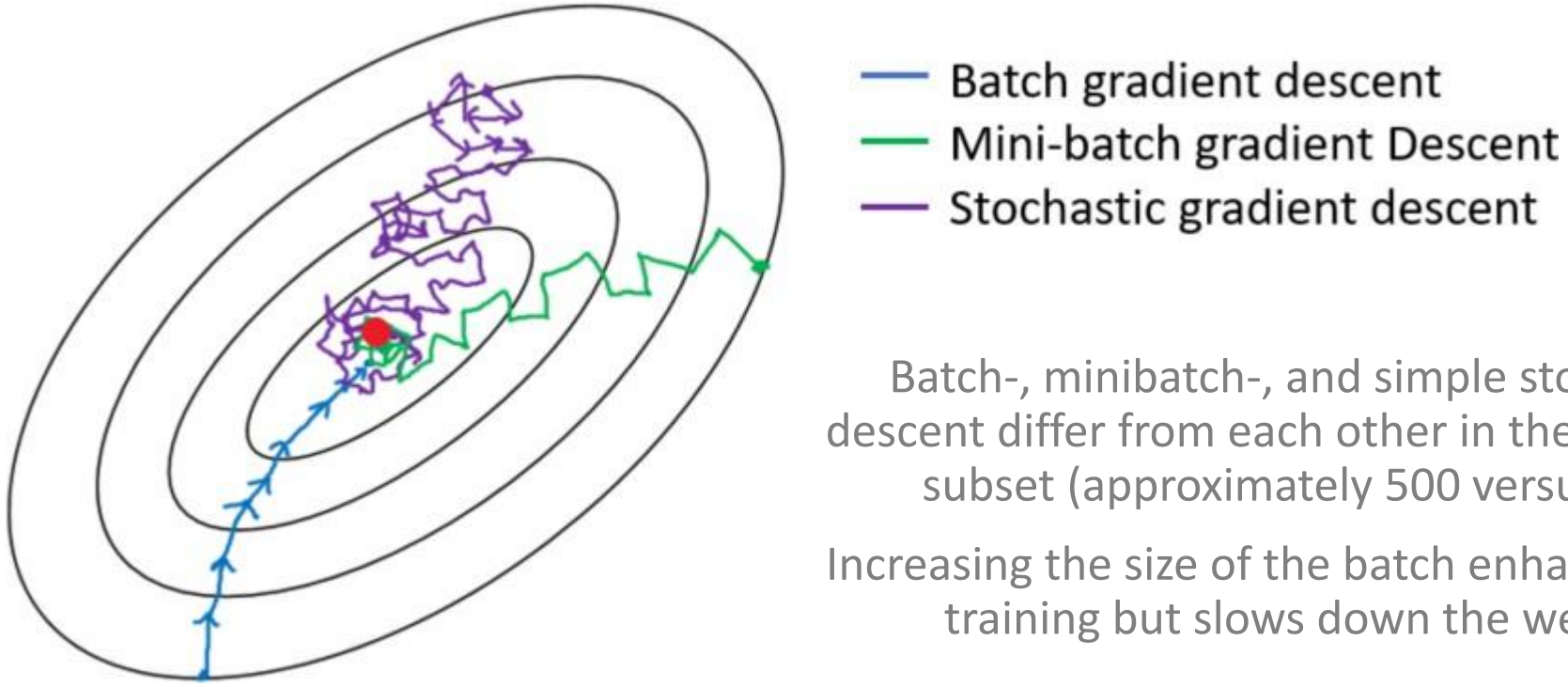
Gradient descent

- There's also stochastic gradient descent....



Gradient descent

- There's also stochastic gradient descent....



Batch-, minibatch-, and simple stochastic gradient descent differ from each other in the size of the random subset (approximately 500 versus 50 versus 1).

Increasing the size of the batch enhances the stability of training but slows down the weight updates.

Gradient descent

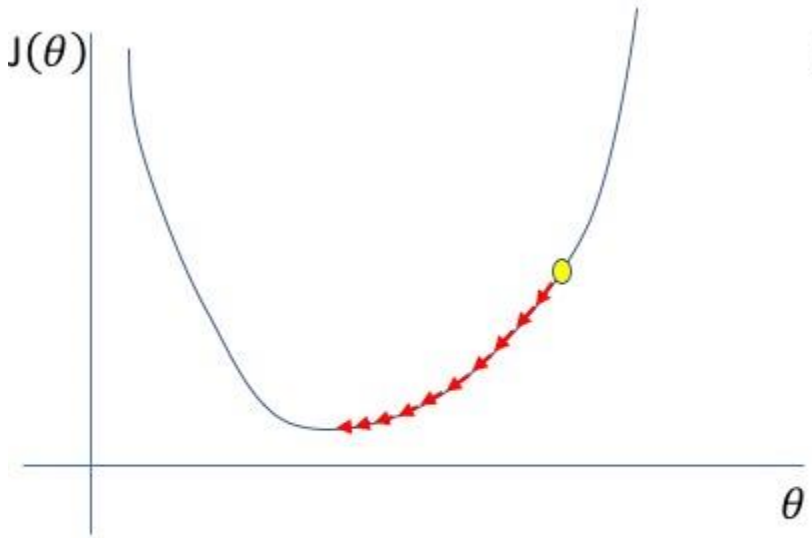
- We also just discussed another problem: how to choose the step size?

Gradient descent

- We also just discussed another problem: how to choose the step size?
- At the beginning of training, the step size should be large to allow gradient descent to converge more quickly to the vicinity of a local optimum.
- Closer to the end of training, the step size should be small to prevent the optimizer from jumping out of the potential well.

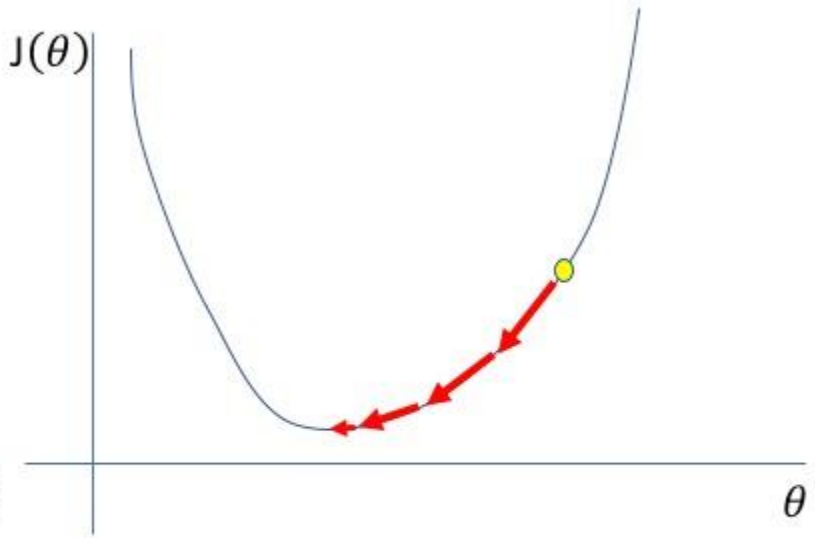
Gradient descent

Too low



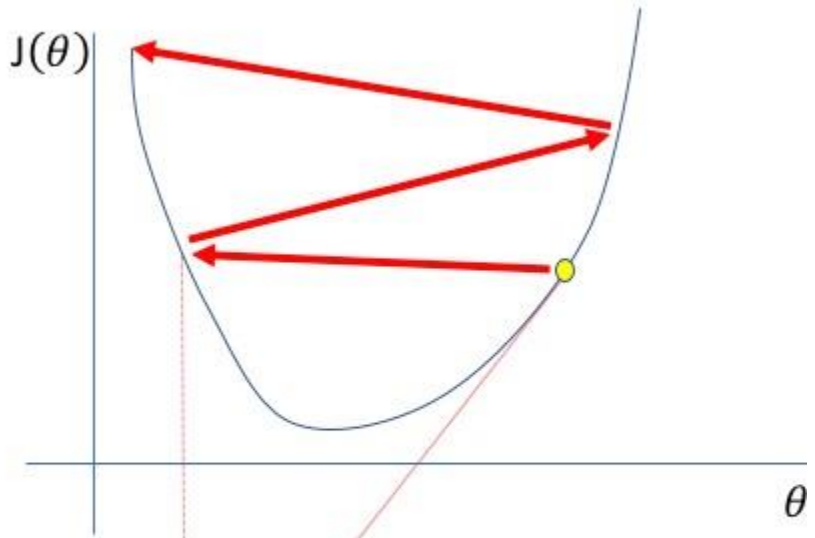
A small learning rate requires many updates before reaching the minimum point

Just right



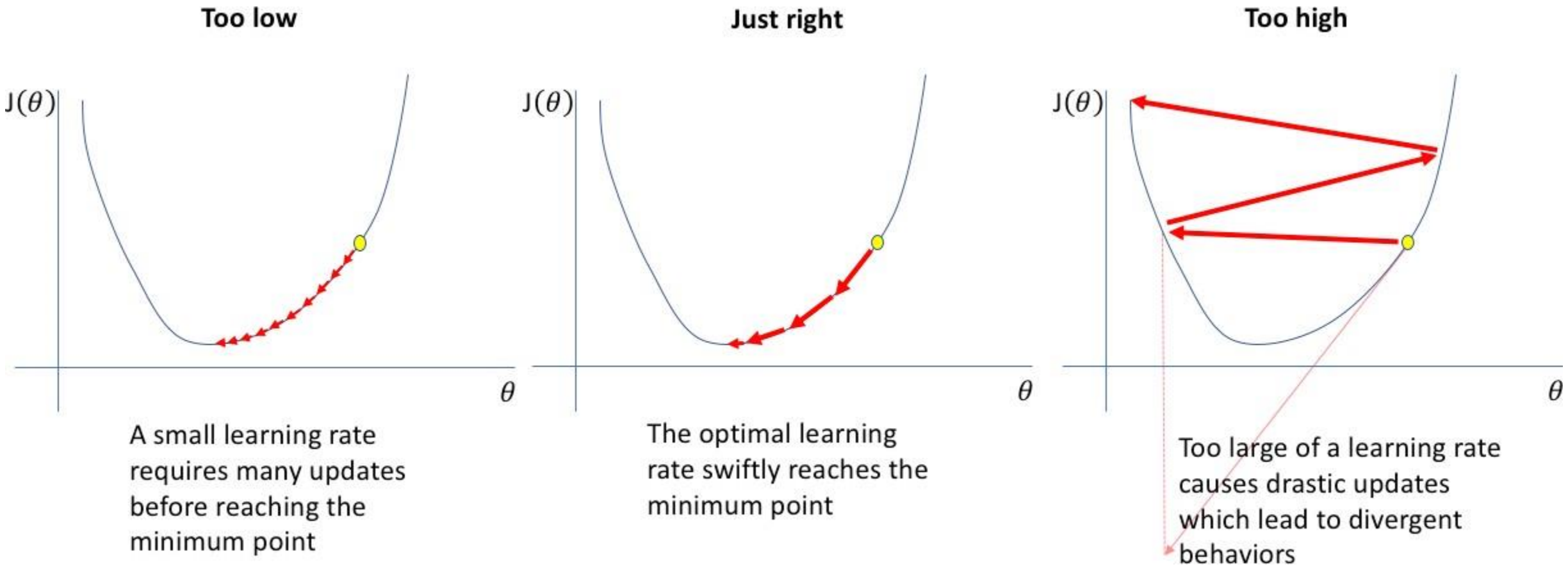
The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

Gradient descent



If the step length is not chosen correctly, the gradient descent will not converge even in the simplest cases.

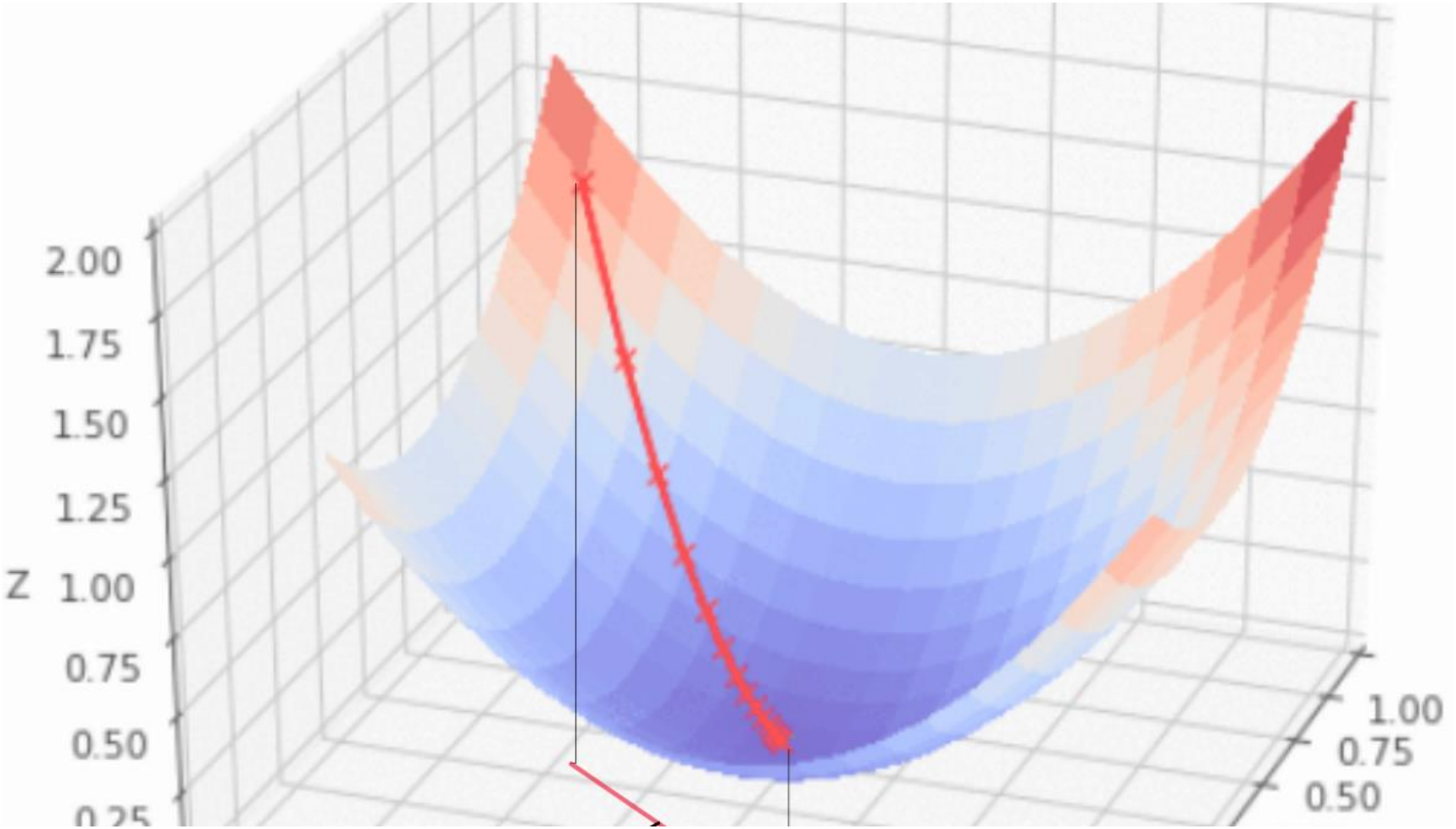
Gradient descent

- Finally, the third problem we discussed is whether gradient descent converges or not.
- How can we determine when our algorithm will be able to find the global optimum?

Gradient descent

- Finally, the third problem we discussed is whether gradient descent converges or not.
- How can we determine when our algorithm will be able to find the global optimum?
- It can do so when the function being optimized is a bounded convex function, and the optimization occurs on a convex set. Such functions have a unique optimum!
- Many important ML error functions such as MSE (Mean Squared Error), MAE (Mean Absolute Error), and others are convex. Typically, the optimization is unconstrained, i.e., it occurs over the entire $\mathbb{R}$^n, which is a convex set.

Gradient descent



In a convex optimization problem, gradient descent with a properly chosen step length will necessarily converge to the global optimum!

Gradient descent

- What can be done if the problem is non-convex?

Gradient descent

- What can be done if the problem is non-convex?
- In such cases, one can run multiple instances of gradient descent in parallel (N gradient descents) and select the best result.
- Additionally, periodic restarts can be performed: resetting the learning rate at an intermediate step to artificially increase the step size. This technique helps to escape from local minima and potentially find better solutions.

Gradient descent

- Indeed, regardless of the problems we have already discussed and partially resolved, we still face at least one significant issue.
- Gradient descent is still slow, even when it's stochastic!

Gradient descent

- Indeed, regardless of the problems we have already discussed and partially resolved, we still face at least one significant issue.
- Gradient descent is still slow, even when it's stochastic!
- In essence—it does not take into account the dynamics of the system: how gradients change, which directions are important, which are not, etc.
- In other words, the algorithm does not take past steps into account. If we can teach it to do so, we can achieve significantly more efficient methods.

Gradient descent

- Indeed, regardless of the problems we have already discussed and partially resolved, we still face at least one significant issue.
- Gradient descent is still slow, even when it's stochastic!
- In essence—it does not take into account the dynamics of the system: how gradients change, which directions are important, which are not, etc.
- In other words, the algorithm does not take past steps into account. If we can teach it to do so, we can achieve significantly more efficient methods.

This will be the subject of our study in future classes :)