Gradient Descent for Risk Optimization

Michael

1 Intuition Behind Gradient Descent

Although the term ‘gradient descent’ is very fancy, the idea and intuition behind this term is extremely simple. It is same for *convergence rate*. All you need to know is that what is the meaning of derivative.

The **derivative** of a function of a real variable measures the sensitivity to change of the function value (output value) with respect to a change in its argument (input value). Let’s draw a table for this definition.

Table 1.1: Derivative

Measures	Input value	Output value
Sensitivity to	Change	Change
↓	↓	↓
	change rate	change rate
	↓	↓
Convergence Rate		

If you got the idea of table 1.1, you can stop reading the section 1 and jump to section 2 and 3. If not, allow me to spend a little more time to explain this.

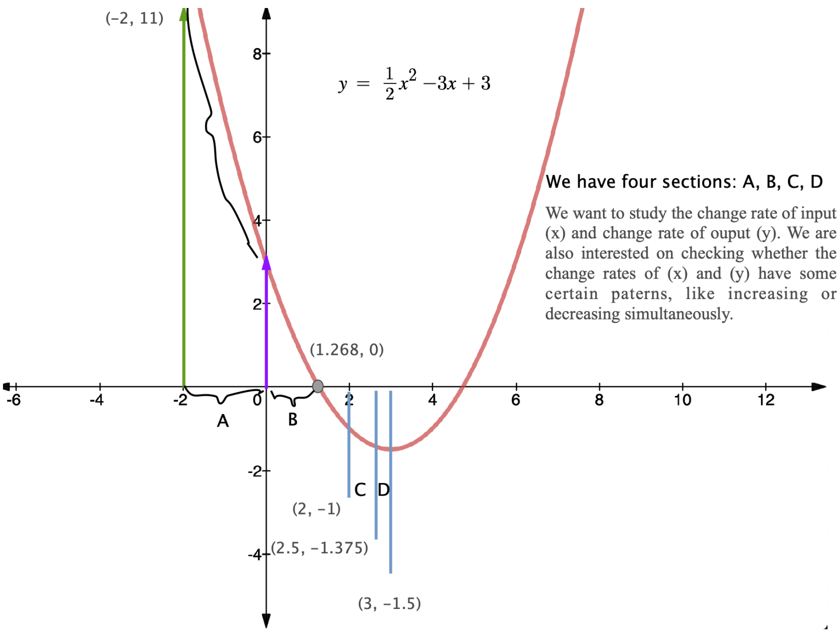


Figure 1.1: Gradient Visualisation

Again, the derivative is not just about the measurement of change of input and output, it also reflects the sensitivity of those changes. Take a look the figure 1.1, tell me where both input (x) and output (y) are changing fast, and where both input (x) and output (y) are changing slowly[1].

From the figure 1.1, we can see that in section A, both input (x) and output (y) are changing quite fast by measuring the differences. When it comes to section C and D, the changing rates for input-x and output-y are decreasing simultaneously. We also say that input x and output y have the same **convergence rate** intuitively[2]. For the **supervised learning** in *machine learning*, it starts to do optimization with the tool of convergence rate.

Now, let's check the formal definition of derivative. The slope m of the secant line is the difference between the y values of these points divided by the difference between the x values, that is,

$$m = \frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{(x+h) - x} = \frac{f(x+h) - f(x)}{h}$$

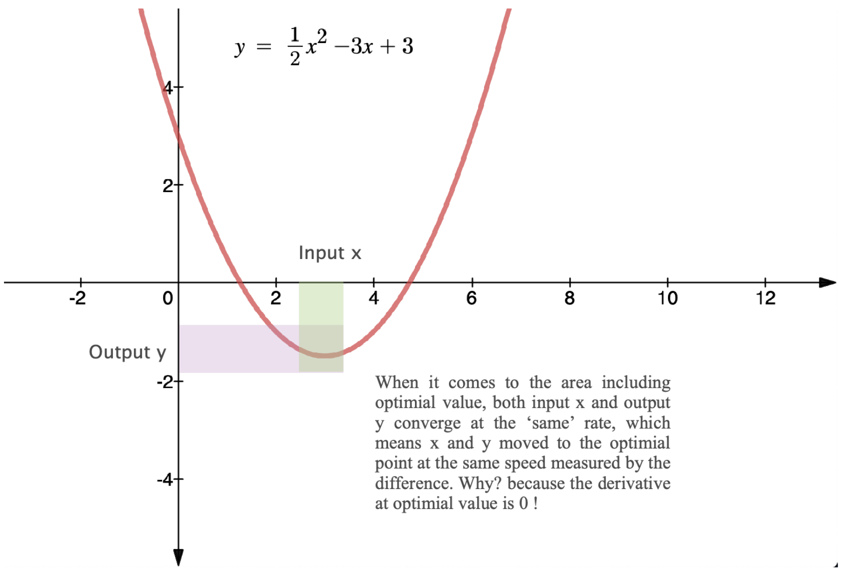
Geometrically, the limit of the secant lines is the tangent line. Therefore, the limit of the difference quotient as h approaches zero, if it exists, should represent the slope of the tangent line to $(x, f(x))$. This limit is defined to be the derivative of the function f at x:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

With the riview of defintion of derivative, we are ready to apply the so called **gradient descent** algorithm to find the minimum of a function.

Definition 1.1. Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function.

Let's disentangle the convergence rate for the function $y = \frac{1}{2}x^2 - 3x + 3$ in figure 1.1 by presenting the following figure and table:



[1]If we assume that speed of changing is measured by the difference between two values.
[2]This is not formal mathematical definition.

Table 1.2: Show the gradient descent step by step

Section	Points	Change of input(x)	Change of ouput (y)	Convergence ratio
A	$(-2, 11)$			
	$(0, 0)$	2	11	$11/2 = 5.5$
B	$(0, 0)$			
	(2,-1)	2	1	$1/2 = 0.5$
C	(2,-1)			
	(2.5,-1.375)	0.5	0.375	$0.375/0.5 = 0.75$
D	(2.5,-1.375)			
	(3,-1.5)	0.5	0.125	$0.125/0.5 = 0.25$
		...		
	(2.9,-1.49499)			
	(2.95,-1.49875)	0.05	0.00376	$0.00376/0.05 = 0.075$
		...		
	(2.98,-1.499799)			
	(2.99,-1.499950)	0.01	0.000151	$0.00376/0.05 = 0.00151$
we have learning rate α to scale the convergence rate = 1				

Markup: it's all about simultaneous moving of input (x) and (y).

2 A Mathematical example

Now, we will employ the gradient descent[3] to find the minimum value for the function in section 1:

$$f(x) = \frac{1}{2}x^2 - 3x + 3$$

The intuition of using gradient descent is that we are trying to iterate the change rate of input x by tracing the change rate of output y. To do this, we need the help from the derivative of the function. Here is the Python code:

```
# Gradient Descent: mathematical example
# @ Michael


# define function
def qudraticfun(x):
    y = 1/2 * x**2 - 3 * x + 3
    return(y)


# define the derivative
def qudraticder(x):
    yprime = x - 3
    return(yprime)


# find the minium numerically
```

[3]Review the definition, it's an algorithm.

```
update_x = 0  # most time we start it from zero
alpha = 0.01  # learning rate
tolerate_rule = 1  # initialize the tolerate rule for stopping iterating
max_iters = 10000  # set the maximum iterative number

i = 0  # interation counting index
while tolerate_rule >= 0.00001 and i <= max_iters:
    start_x = update_x  # set the starting value
    update_x = start_x - alpha * qudraticder(start_x)
    tolerate_rule = abs(update_x - start_x)

print("The minimum value is", qudraticfun(update_x),
      "when x is equal to", update_x)

# The minimum value is -1.4999995136117308 when x is equal to
  2.999013705653870
```

3 A Machine Learning Example

References