AIX20006 Assignment - Gradient Descent Algorithm

Class #01

Prof. Kim JungHyun

Team #11 (22100748 Choi Yunyoung, 22200316 Park Jueun)

*Spring 2023*

**AIX20006 Assignment – Gradient Descent Algorithm**

**Due date**

Friday, March 31st, 2023.   
Please note that no late submissions will be accepted.

**Assignment description**

The objective of this assignment is to exercise the implementation of the gradient descent algorithm. The simulation code should be written in Python. The main deliverables for this assignment are 1) a report documenting your efforts on this assignment and 2) simulation codes you developed. All deliverables must be a joint effort, meaning that your team will need to distribute the different tasks amongst the members to share the workload fairly and effectively. However, please note that one student from your team should submit materials on behalf of the entire team.

**Team**

Students will work in a team of two or three students with the aim of developing teamwork skills while working with other students. The entire team will receive the same base grade; however, I will ask each of you to submit your own peer assessment in which you evaluate the contributions of your teammates; therefore, the grade may be changed as needed. The purpose of this assessment is to find ways to work well together and contribute equally to the overall product.

**Problem statement**

Suppose that you are asked to find the minimum point of the objective function within −5 ≤ x ≤ 5:

f(x) = sinx − cosx

Deliverables for this problem are as follows:

✈ Create a plot showing the objective function within −5 ≤ x ≤ 5.

✈ Write a pseudo-code for your own algorithm.

✈ Explain how you design a stopping criterion.

✈ Discuss with your teammates and answer the following question:

➤ Suppose that you implement a stopping criterion where the algorithm terminates when

the number of iterations exceeds the maximum number of iterations. When the following

conditions are given, what is the optimal solution? Do you think that it is a global

optimum within the pre-determined range? If not, what is your strategy to improve the

fidelity of the solution provided by the algorithm?

✔ Initial point x = 3

✔ Step size = 0.05

✔ The maximum number of iterations = 500

➤ Suppose that you implement a stopping criterion where the algorithm terminates when

the number of iterations exceeds the maximum number of iterations. When the following

conditions are given, do you think that the algorithm provides a global optimum within

the pre-determined range? If not, what is your strategy to improve the fidelity of the

solution provided by the algorithm? You could set up a different initial point; however, it

is assumed that the initial point cannot be changed. You are only allowed to control either

step size or the maximum number of iterations. Which option would like to choose?

What is the reason why you end up choosing the option?

✔ Initial point x = 2

✔ Step size = 0.01

✔ The maximum number of iterations = 50

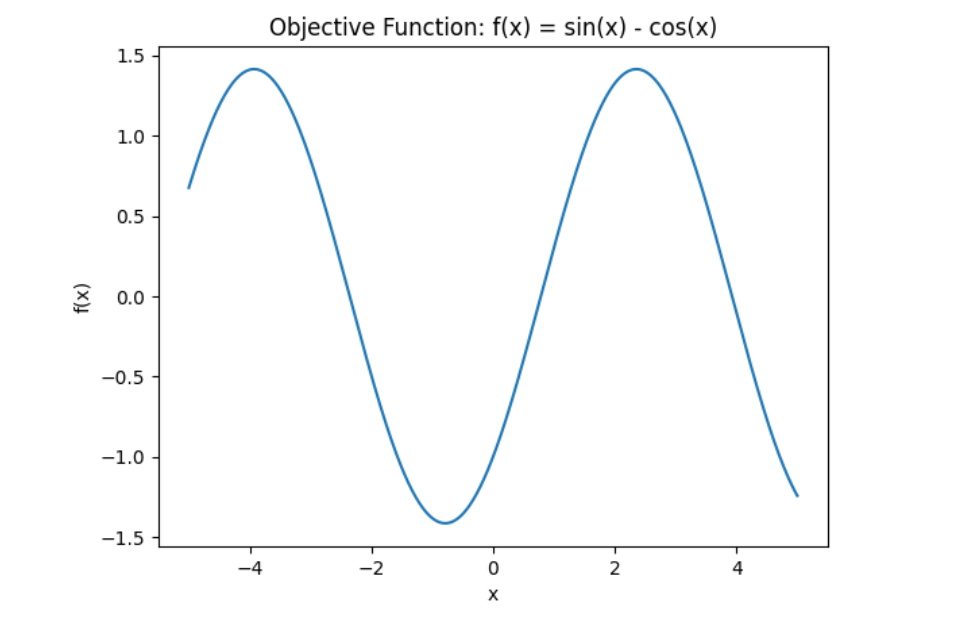
Please note:

It is possible to solve this problem with various built-in libraries; however, this is not the point. You are asked to write your own gradient descent algorithm code based on what you have learned in the class.

**Citation**

Please ensure that all codes and materials originating from other sources including ChatGPT must be clearly documented.

✈ Create a plot showing the objective function within −5 ≤ x ≤ 5.



✈ Write a pseudo-code for your own algorithm.

| Pseudo-code:  Input: initial\_point, step\_size, max\_iter, function(sinx - cosx)  Output: the learned model parameters, plot  Start  x = intial\_point of x  f(x) = function(sinx - cosx)  Define the initial point: x = ⬜  step\_size = ⬜  max\_iter = ⬜  for (start = 0, end = max\_iter):   1. compute the gradient of the loss function   gradient = f’(x)   1. stop when x meets stopping criterion   If (the absolute value of the gradient) ≤ le-6 or (the absolute value of the gradient) ≤ 0:  exit the for loop   1. initialize the initial point and function as a list type.   x\_list = [x]  y\_list = [f(x)]   1. update the model parameters   for i in range(max\_iter):  x = x - (step\_size \* gradient)  update x\_list and y\_list  if (stopping criterion):  Break  return x\_list, y\_list  Compute the gradient descent path  - assign the following gradient(x, step\_size, max\_iter) values into x\_list and y\_list.  Decide a range of x and y values to plot  - x value starts from -5 to 5 with 500 equality space.  - y value contains the value of f(x) for each value in x values.  Create the plot  - plot a graph with x values on x-axis and y values on the y-axis.  - red line graph is plotted with circles at each data point.  Label the title  - title with “Objective Function: f(x) = sin(x) - cos(x)”  Display the plot  End |
| --- |

✈ Explain how you design a stopping criterion.

| Initial point:  Initial point should be in a range of -3.92 ~ 2.36 to get the global minimum.  Maximum number of iterations:  - If the step size is big enough(between 0.1 ~ 0.5) to reach the global minimum, a maximum of 50 iterations would be sufficient. We empirically found those numbers.  - If the step size is small(less than 0.1) to reach the minimum, the maximum number of iterations should not be more than 500 because it takes quite a long time. If we haven't found the minimum after 500 iterations, then we should consider changing the step size or other factors instead of the maximum number of iterations.  Gradient of f(x):  If the gradient of f(x) is less than 1e-6, stop the process. We don't need to find it anymore because it's close enough to zero. |
| --- |

✈ Discuss with your teammates and answer the following question:

➤ Suppose that you implement a stopping criterion where the algorithm terminates when

the number of iterations exceeds the maximum number of iterations. When the following

conditions are given, what is the optimal solution? Do you think that it is a global

optimum within the pre-determined range? If not, what is your strategy to improve the

fidelity of the solution provided by the algorithm?

✔ Initial point x = 3

✔ Step size = 0.05

✔ The maximum number of iterations = 500

| Answer:  Optimal solution is an act, process, or methodology of making something as fully perfect, functional, or effective as possible. According to the definition, we suppose that x = 5 is the optimal solution within the pre-determined range, the lowest point possible. However, it is not a global optimum.    - *Graph of given conditions* -  The green line on this graph is the global minimum, and the blue line which is the minimum found by the model under that condition, has a larger value than the green line. So we know that the value the model found is not the global minimum within the pre-determined range.  Strategy:  Control the initial point(as we know the coordinate point of the minimum point). If we change the initial point to a range between two maximum points (-3.92 ~ 2.35), we can get the minimum point of the objective function. When the initial point is either below -3.92 or above 2.34, plots will iterate into a opposite way from the minimum point.      With the same step\_size and max\_iter, we could see the model found the global minimum by just moving the initial point. |
| --- |

➤ Suppose that you implement a stopping criterion where the algorithm terminates when

the number of iterations exceeds the maximum number of iterations. When the following

conditions are given, do you think that the algorithm provides a global optimum within

the pre-determined range? If not, what is your strategy to improve the fidelity of the

solution provided by the algorithm? You could set up a different initial point; however, it

is assumed that the initial point cannot be changed. You are only allowed to control either

step size or the maximum number of iterations. Which option would like to choose?

What is the reason why you end up choosing the option?

✔ Initial point x = 2

✔ Step size = 0.01

✔ The maximum number of iterations = 50

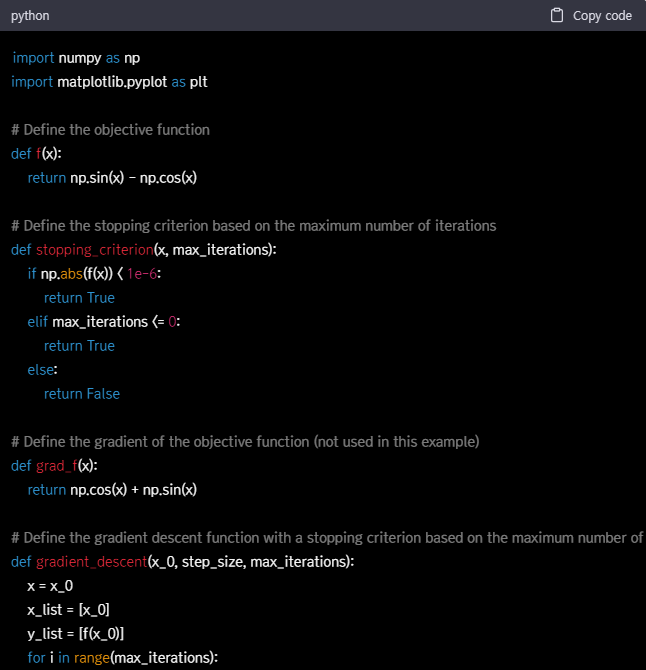
| Answer:  No. algorithm doesn’t provide a global optimum.    - *Graph of given conditions* -  Due to the small step size and the number of iterations, the global minimum cannot be found even though it is an appropriate initial point.  Strategy:  Change a step size. Either changing step size or maximum number of iterations let us able to know the minimum point of the plot. Yet, if the number of repetitions is extremely optimized, it will be difficult to find the optimum point. So it is more efficient to change the step size. Also by using %%timeit command, we could know that changing the step size took less time than changing the number of iterations.  **We decided to change(increasing) a step size: 0.01 → 0.1**      **Change max\_iteration: 50 → 500** |
| --- |

**Citation**

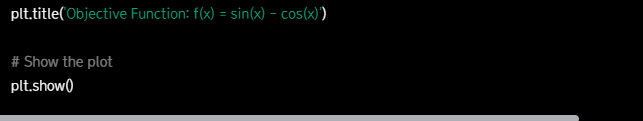
#<https://chat.openai.com/chat/08bd6728-a208-4ca4-91d1-46bab13672f9> - ChatGPT

* plot coding, psuedocode draft

- *Plot coding* -







- *Pseudocode draft* -

