

A3 (16%) Submission due on 14th November 2025 (Week 13 Friday)

Write a single python file to perform the following tasks:

- (a) Perform gradient descent to minimize the cost function $f_1(a) = a^5$ with an initialization of $a = 2.5$
- (b) Perform gradient descent to minimize the cost function $f_2(b) = \sin^2(b)$ with an initialization of $b = 0.5$ (where b is assumed to be in radians)
- (c) Perform gradient descent to minimize the cost function $f_3(c, d) = c^3 + d^2 \sin(d)$ with an initialization of $(c, d) = (2, 4)$. (where c, d are assumed to be in radians)

Submit a single python file with filename “A3_StudentMatriculationNumber.py”. It should contain a function A3_MatricNumber that takes the following inputs and returns the following outputs in the following order:

Python function inputs:

- learning_rate: A fraction that falls between 0 and 0.2 (e.g., 0.1, etc.)
- num_iters: A positive integer specifying the number of gradient descent iterations for each minimization task. Note that for the purpose of this exercise, you do not need to check for convergence. Just simply run the gradient descent algorithm for num_iters iterations.

Python function outputs in the following order:

- a_out: numpy array of length num_iters containing the updated values of a. For example, a_out[0] is the value of a after the first round of gradient descent (NOT the initialized value of a). (2%)
- f1_out: numpy array of length num_iters containing the updated values of f1. For example, f1_out[0] is the value of f1 after the first round of gradient descent, i.e., the value of f1 given a_out[0]. (2%)
- b_out: numpy array of length num_iters containing the updated values of b. For example, b_out[0] is the value of b after the first round of gradient descent (NOT the initialized value of b). (2%)
- f2_out: numpy array of length num_iters containing the updated values of f2. For example, f2_out[0] is the value of f2 after the first round of gradient descent, i.e., the value of f2 given b_out[0]. (2%)
- c_out: numpy array of length num_iters containing the updated values of c. For example, c_out[0] is the value of c after the first round of gradient descent (NOT the initialized value of c). (2%)
- d_out: numpy array of length num_iters containing the updated values of d. For example, d_out[0] is the value of d after the first round of gradient descent (NOT the initialized value of d). (2%)
- f3_out: numpy array of length num_iters containing the updated values of f3. For example, f3_out[0] is the value of f3 after the first round of gradient descent, i.e., the value of f3 given c_out[0] and d_out[0] (4%)

Please use the python template provided to you. Remember to rename both “A3_StudentMatriculationNumber.py” and “A3_MatricNumber” using your student matriculation number. For example, if your matriculation ID is A1234567R, then you should submit “A3_A1234567R.py” that contains the function “A3_A1234567R”. Please do NOT zip/compress your file. Because of the large class size, **points will be deducted if instructions are not followed**. The way we would run your code might be something like this:

```
>> import A3_A1234567R as grading
>> learning_rate = 0.1
>> num_iters = 10
>> a_out, f1_out, b_out, f2_out, c_out, d_out, f3_out =
grading.A3_A1234567R(learning_rate, num_iters)
```

Submission folder: Canvas/EE2211/Assignments/A3

(The submission folder in Canvas will be closed at 23:59 on Friday of Week 13 sharp. No extensions will be entertained.)