

OMML 19-20

Homework 1: Hands on Python

Marco Boresta

1 October 2019

1 Some exercises

Here is a list of some exercises. Those of you that already have experience in coding will find them easy. The others might struggle a little with some of them. Try not to stop at the first difficulties!

- 1). Create an array with shape (10,4) of random numbers.
- 2). Create an array with shape (8,1) of random numbers whose value is between 3 and 9.
- 3). Create an array with shape (20,2) of integer random numbers whose value is between 1 and 10.
- 4). Create an array of random numbers that can be multiplied with the array created in the point 3) to obtain a resulting array with 20 rows and 1 column. Compute then the product and be sure that the shape obtained is the one required.
- 5). Take what obtained at point 4) and apply to it a sigmoid function. (A what?! Ask Google!)
- 6) Define a function that takes in input a (3,1) array and returns the cube of its first element plus the square root of the second minus three times the third.

- 7) Define also a function that computes its gradient and check what they return with an input you choose.

1.1 Other exercises

- 8) Reversing an array can be a tough task, especially for a novice programmer. Mary just started coding, so she would like to start with something basic at first. Instead of reversing the array entirely, she wants to swap just its first and last elements.
Define a function *firstReverseTry* that takes in input an array *arr*, swaps its first and last elements and returns the resulting array.

Example:

For *arr* = [1, 2, 3, 4, 5], the output should be:

firstReverseTry(*arr*) = [5, 2, 3, 4, 1].

For *arr* = [-1, 9], the output should be:

firstReverseTry(*arr*) = [9, -1].

- 9) Define a function *sameElementsNaive* that takes in input two lists and finds the number of elements that are contained in both of the given lists.

Example:

For *a* = [1, 2, 3] and *b* = [3, 4, 5], the output should be:

sameElementsNaive(*a*, *b*) = 1.

For *a* = [1, 2, 3] and *b* = [2, 3, 4], the output should be:

sameElementsNaive(*a*, *b*) = 2.

- 10) Define a function *digitsProduct* that takes in input an integer *product* and finds the smallest positive (i.e. greater than 0) integer the product of whose digits is equal to *product*. If there is no such integer, return -1 instead.

Example:

For *product* = 12, the output should be:

digitsProduct(*product*) = 26;

For *product* = 19, the output should be:

`digitsProduct(product) = -1.`

For `product = 576`, the output should be:

`digitsProduct(product) = 889.`

- 11) Define a function *removeDuplicateCharacters* that takes in input a string `str`, and returns the string obtained after removing all characters from `str` that appear more than once in it.

Example:

For `str = "zaabcbcd"`, the output should be:

`removeDuplicateCharacters(str) = "zcd"`.

For `str = "zzzzzzz"`, the output should be:

`removeDuplicateCharacters(str) = ""`.

For `str = "neuralnetwork"`, the output should be:

`removeDuplicateCharacters(str) = "ualtwok"`.