Assignment #1 - Surgical Data Science

Context

This assignment is an opportunity to demonstrate your knowledge and practice solving problems about artificial neurons and gradient descent. This assignment is implementation-based (i.e. you will be asked to write code to solve one or more problems).

Logistics

Assignment due date: 2026-01-28

Assignments are to be submitted electronically through Brightspace. It is your responsibility to ensure that your assignment is submitted properly and that all the files for the assignment are included. Copying of assignments is NOT allowed. High-level discussion of assignment work with others is acceptable, but each individual or small group is expected to do the work themselves.

Implementation Part

Programming language: Python 3

For all parts of the implementation, you may use the Python Standard Library (https://docs.python.org/3/library/). Unless explicitly indicated below or explicitly approved by the instructor, you may not use any additional packages.

You must implement your code yourself; do not copy-and-paste code from other sources. Please ensure your implementation follows the specifications provided; it may be tested on several different test cases for correctness. Please make sure your code is readable; it may also be manually assessed for correctness. You do not need to prove correctness of your implementation.

You must submit your implementation as a single file named "assignment1.py" with functions as described below (you may have other variables/functions/classes in your file). Attached is skeleton code indicating the format your implementation should take.

Experiments Part

You must submit your report on experimental results as a single file named "assignment1.pdf".

Implementation Part

Question 1 [10 marks]

Write a function that simulates an artificial neuron. The neuron should use an aggregation function that is a weighted sum of inputs. The neuron should use the signmoid linear unit (SiLU) as the activation function. Do not include a bias term in the neuron.

$$SiLU(x) = \frac{x}{1 + \exp(-x)}$$

The function must be named "artificial_neuron".

The function should take two input arguments: (1) a list of inputs x of length n, and (2) a list of weights w of length n.

The function should return one value: (1) the output of the neuron.

Question 2 [10 marks]

Write a function that performs standard gradient descent on a multi-variable function f, to estimate the minimum of the multi-variable function (note that your function might not find the global minimum). Your function may assume that the gradient ∇f is provided.

The function must be named "gradient_descent".

The function should take four input arguments: (1) the multi-variable function f whose value may be computed by calling it with a list of coordinates of length n (i.e. a callback function), (2) ∇f, the gradient of f, whose value may be computed by calling it with a list of coordinates of length n (i.e. a callback function), (3) a list of coordinates of length n, indicating an initial guess for the minimum on f, (4) the learning rate α.

The function should return two values: (1) a list of coordinates of length n, indicating the minimum that was found, and (2) the value of f at the minimum that was found.

Question 3 [10 marks]

Write a function that creates a neural network in PyTorch by subclassing the "torch.nn.Module" class in PyTorch. The model must have at least one linear layer. The model may have any architecture, provided it has at least one linear layer.

The function must be named “pytorch_module”.

The function should take no input arguments.

The function should return one value: (1) a PyTorch module object.

For this question, you may use the PyTorch library and any other libraries it depends on.

Experiments Part

Question 4 [10 marks]

a) Consider the following function of two variables a, b. Assume x_1, x_2, y_1, y_2 are constants.

$$f(a,b) = \frac{1}{2}((ax_1 + b - y_1)^2 + (ax_2 + b - y_2)^2)$$

Use your implementation of gradient descent from above to compute the minimum value of the function f with respect to the parameters a, b when the constants have values $x_1 = 3, x_2 = -2, y_1 = 0.5, y_2 = -0.75$. In your answer, state the gradient of f, the values of a, b at which the minimum is achieved, the minimum value for f, the initial guess used to find the minimum, and the learning rate used.

b) Consider the following function of two variables a, b. Assume x_1, x_2, y_1, y_2 are constants.

$$f(a,b) = \frac{1}{2}((SiLU(ax_1 + b) - y_1)^2 + (SiLU(ax_2 + b) - y_2)^2)$$

Use your implementation of gradient descent from above to compute the minimum value of the function f with respect to the parameters a, b when the constants have values $x_1 = 3, x_2 = -2, y_1 = 0.5, y_2 = -0.75$. In your answer, state the gradient of f, the values of a, b at which the minimum is achieved, the minimum value for f, the initial guess used to find the minimum, and the learning rate used.