EXERCISE 1

DR. VICTOR UC CETINA

1. Stochastic Gradient Descent

(1) Generate 100 artificial data points (x_i, y_i) where each x_i is randomly generated from the interval $[0, 1]$ and $y_i = \sin(2\pi x_i) + \varepsilon$. Here, ε is a random noise value in the interval $[-0.3, 0.3]$.

(2) Implement in your favorite programming language the Stochastic Gradient Descent algorithm to solve the regression problem using the 100 data points you generated.

Loop {

for $i = 1$ to m {

$$\theta_j := \theta_j + \alpha\big[y_i - h_\theta(x_i)\big](x_i)_j \qquad \text{(for every } j\text{)}.$$

}

}

Where:

- i is an index defined over the number of data points, from $i = 1$ to $m = 100$.
- j is an index defined over the terms of the polynomial, from $j = 0$ to $j = D$.
- The last factor $(x_i)_j$ means: the factor multiplying parameter θ_j in the polynmial function, which in this case it will be x_i to the power of j.

Note: the use of machine learning libraries such as scikit-learn is forbidden.

(3) Make your initial learning rate constant $\alpha = 0.001$, and train a polynomial model using your artificially created data. A polynomial model has the form $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \ldots + \theta_D x^D$, where D is the degree of the polynomial.

(4) All initial θ_i parameters are randomly generated in the interval $[-0.5, 0.5]$.

(5) Try different values for D.

(6) Try different α values to speed up the learning process.

2. Report Submission

- Deadline: Feb 5th, 2026.
- Upload your code and report to your personal GitHub repository.

3. Report

Make sure to include the following:

(1) Your final model (polynomial fuction) with optimal learned parameters.

(2) Your final α value.

(3) One graph containing the cloud of the training data points, the sine function, and the learned polynomial function. Please use different colors for each one.

(4) A second graph showing the error curve. It should clearly illustrate how the error of your model decreases as the number of iterations is increased. For each combination of α, D, θ we can evaluate the error (performance) of the model using the root-mean-square error E_{RMS}:

$$E_{\text{RMS}} = \sqrt{2E(\theta)/m}$$

where

$$E(\theta) = \frac{1}{2}\sum_{i=1}^{m}\{h_\theta(x_i) - y_i\}^2$$