1ST EXERCISE SESSION

Exercise 1: supervised learning, unsupervised learning and reinforcement learning

Classify the following learning problems as supervised learning, unsupervised learning and reinforcement learning tasks.

(a) Identification of products frequently bought together
(b) Chess computer capable of learning from previous games
(c) Spam recognition and filtering
(d) Classification of applicants as credit-worthy or unworthy
(e) Object recognition in computer vision
(f) Obstacle avoidance in robotics
(g) Automatic sorting of images wrt the depicted objects

Exercise 2: Data preparation and preprocessing

Data preprocessing is an integral step in A.I problems as the quality of data and the useful information that can be derived from it directly affects the ability of A.I models to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model. We will practice on a dataset related to breast cancer tumor classification available on:
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29
Each row of the data contains the Id, 9 features and 1 label (class)
We want to prepare this dataset for a classification model that uses the 9 features to predict the class of the cancer tumor.

a) Download and upload the data set breast-cancer-wisconsin.data.txt enclosed with this exercise
b) Drop the missing or non-numeric values
c) Drop the ID column
d) Create features and labels arrays X and y
e) Scale/Standardize the features array X (use Minmax or Standard scaling)
f) Transform the feature array y to a binary array 0 or 1
g) Split the arrays into training and test arrays (4 resulting arrays)

Exercise 3: Gradient descent from scratch

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, gradient descent is an optimization technique used for computing the model parameters (coefficients and bias) for algorithms like linear regression, logistic regression, neural networks, etc. In this technique, we repeatedly iterate through the training set and update the model parameters in accordance with the gradient of error with respect to the training set.
Consider a model g characterized by two parameters m and b.
$g(x)=mx+b$
We want to approximate an unknown function y characterized by N tuples

$$(xi, yi), i \in \{1, \dots N\}$$

Given the cost function:

$$f(m,b)=\frac{1}{N}\sum_{i=1}^{N}(yi-(mxi+b))^2$$

The gradient can be calculated as:

$$f'(m,b)=\begin{bmatrix}\frac{df}{dm}\\ \frac{df}{db}\end{bmatrix}=\begin{bmatrix}\frac{1}{N}\sum -2x_i(y_i-(mx_i+b))\\ \frac{1}{N}\sum -2(y_i-(mx_i+b))\end{bmatrix}$$

And the parameters update at each iteration is performed as:

$$m_{t+1}=m_t-\frac{df}{dm}*learning_{rate}$$

$$b_{t+1}=b_t-\frac{df}{db}*learning_{rate}$$

Task: Implement a gradient descent algorithm and approximate a given dataset

a) Load the data in GD_Example.txt (it includes 500 pairs of data (xi,yi))
b) Plot the data points
c) Implement the cost function
d) Using gradient descent algorithm with 500 iterations, find the best fitting line characterized by: $mx+b$. (determine m and b)
e) Plot the final fitting line alongside with the scattered data points.