Introduction to Artificial Intelligence – summer 2022 – exercise #1

Teacher: mgr inż. Maciej Szymkowski, maciej.szymkowski@pw.edu.pl

Please be aware that this exercise is done in pairs (you should inform us about the groups via Teams #General channel – under the post of prof. Paweł Wawrzyński).

Assignment description

In this exercise you will implement two methods – Gradient Descent method and Newton's method for function minimalization (please, be aware that the user should have a possibility to select which method will be taken for function minimalization).

Your program should have a possibility to optimize two types of functions - $F(x) = ax^3 + bx^2 + cx + d$ (where a, b, c, d are scalar numbers – all parameters are specified by the user) and $G(x) = c + b^T x + x^T Ax$ (where c is a scalar number, b is a d-dimensional vector, and A is a positive-definite matrix – all parameters are specified by the user).

The result of your solution is found solution x^* and function value $F(x^*)\ or\ G(x^*)$. It should be possible to define starting point for the optimization procedure in two ways: either user can directly provide initial vector (or scalar number in the case of function $F(x)$) or its elements get generated based on drawing numbers (one number in the case of function $F(x)$) from a uniform distribution defined for the range $[low, high]$ (the range is defined by the user).

In the case of stopping conditions, your program should provide three possibilities:

- Maximum number of iterations
- Desired value $F(x)$ or $G(x)$ to reach (so the process is finished when $F(x) \geq value\_to\_reach$ / $G(x) \geq value\_to\_reach$)
- Maximum computation time

Moreover, your program must provide batch/restart mode. It means that the optimization process will be restarted *n*-times (n-defined by the user). In this mode, your program needs to calculate *n*-times values x^* and function value $F(x^*)\ or\ G(x^*)$, then as the output mean values and standard deviation are reported. This mode should work exactly in the same manner as manually (and independently) running the optimization process for the same $F(x)\ or\ G(x)$, function multiple times. So, if the starting point is set for random generation, it changes for every run.

Technical requirements

Please, be aware that all logic needs to be implemented directly by you. **It is not allowed to use any ready-to-use machine learning or optimization libraries / packages (e.g., PyTorch, TensorFlow).** It means that the task is to implement whole logic by yourself (of course, you can use basic linear algebra and math processing packages / libraries – e.g., NumPy).

The preferred language to solve this task is Python. However, it is still possible to prepare the solution with Java/Scala/C/C++/R/C# languages.

How-to submit the task?

- **Deadline:** You should submit the task (source code of your solution) to maciej.szymkowski@pw.edu.pl not later than 09.03.2022 (Wednesday), 11:59:59 AM CET
- In the title of your e-mail please include "[EARIN] Exercise #1". Moreover, do not forget to include names, surnames, and e-mails of all members of your group (please add them in the e-mail content).

- The results will be discussed during next meeting. The details related to it will be given within our Microsoft Teams channel.
- You can get 0-5 points for this task.
- If you have any questions, please do not hesitate to contact me via e-mail (maciej.szymkowski@pw.edu.pl) or directly by Microsoft Teams platform.