ASSIGNMENT 1: PART 1 [30 MARKS]
MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

This is an individual assignment.

The quiz will be given on September 08, 2023 during the class. See the next page for the homework component which is due September 19, 2023, 11.59 pm Phoenix time.

ASSIGNMENT 1: PART 2 [70 MARKS]
MULTI-OBJECTIVE BAYESIAN OPTIMIZATION

This is an individual assignment. Submit before September 19, 2023, 11.59pm Phoenix time.

Problem:

Bayesian optimization can be used to find the minimum of two black box functions f_1 and f_2 with the help of the BoTorch library. In this assignment, the two black box functions are neural networks. Assume that querying the neural network is possible but very expensive. Therefore, we have to find the minimum by taking as less samples as possible. If an oracle tells us the shape of the black box functions, they will look somewhat similar to the plots shown in Figure 1.

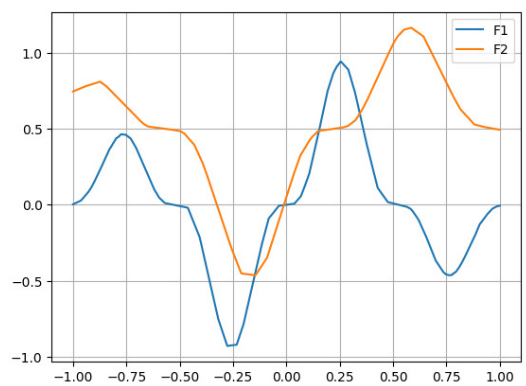


Figure 1: Plots of the black box functions f_1 and f_2 on $[-1, 1]$.

Instructions:

1. **Getting to know software libraries**. Open the *.ipynb notebook in Google Colab. Install BoTorch v.0.9.2 in Python 3 using the command `!pip install botorch`. Do not use software packages other than numpy, matplotlib, pytorch, botorch, and warnings without teaching team's prior approval. [5 marks]

2. **Examining the black box functions**. Load the black box functions f_1 and f_2 defined by the neural networks. This can be done by loading the corresponding neural network weights from "net_f1.pth" and "net_f2.pth." You may call the function f_1 and f_2 every time you need to get the values of $f_1(x)$ and $f_2(x)$ for $x \in [-1, 1]$. In order to understand the task, plot the black box functions f_1 and f_2 for $x \in [-1, 1]$. Note that you will not be able to plot these functions in real applications. [5 marks]

3. **Setting up the Gaussian process and acquisition function**. Define a python function that serves as your main Bayesian optimization loop for multi-objective optimization. The input to the function should be all the input points x you have at the current iteration with their values from objective functions f_1 and f_2. The output of the function should be the next input point x obtained in Bayesian optimization. You can implement your Bayesian optimization function:

   (i) Define the Gaussian process model for each objective given all the input points x and their values from objective function f_1 and f_2. Then, maximize their marginal log likelihoods. [5 marks]

   (ii) Construct the multi-objective acquisition function using `qExpectedHypervolumeImprovement()` [1] provided by BoTorch for multi-objective Bayesian optimization. [5 marks]

[1] `https://botorch.org/api/acquisition.html`

(iii) Optimize the acquisition to find the next candidate x. [5 marks]

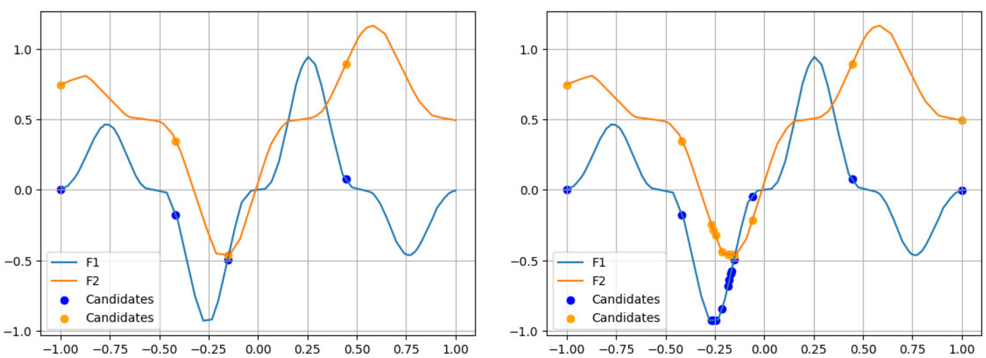
4. **Running Bayesian optimization**. With the main Bayesian optimization loop function, you can now run Bayesian Optimization.

(i) You can first generate initial input points x and calculate their values for the two objectives. [5 marks]

(ii) Start with three x values randomly sampled within the range of $[-1, 1]$. [5 marks]

(iii) Next, iteratively run your Bayesian optimization function on current training data to find new candidate, and add the new candidate to your training data for the next iteration of Bayesian optimization. You can run the Bayesian optimization for 15 iterations in this assignment. [15 marks]

(iv) You should print the candidate x obtained at each iteration. [5 marks]
You should also plot all the candidates and their values of the two objectives at each iteration, similar to Figure 2. [5 marks]



(a) 1-st iteration (b) 10-th iteration

Figure 2: Plots of all points x after the 1-st and 10-th iteration iteration.

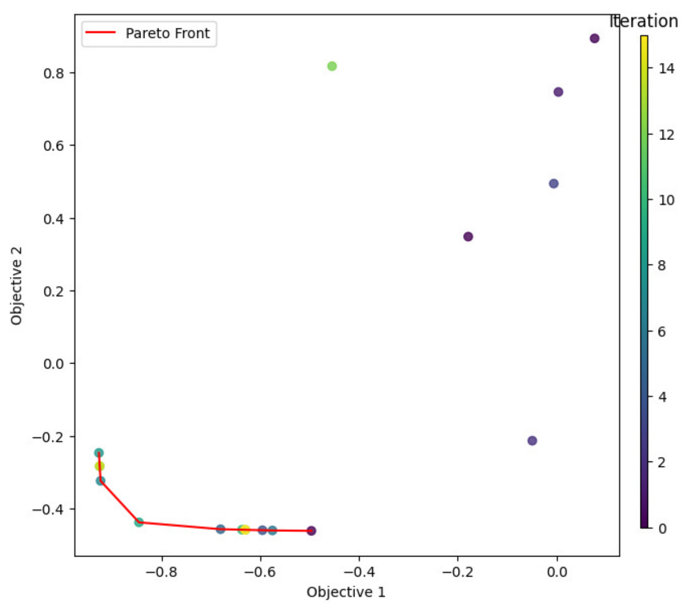


Figure 3: Plots of objective function f_1 vs. f_2 on $[-1, 1]$.

5. **Examining the Pareto front**. After 15 iterations, plot the values of objectives f_1 vs. f_2 for all candidates and mark the Pareto front, similar to Figure 3. [10 marks]

Submissions:

Please finish the assignment in the *.ipynb notebook provided on Google Colab. The final notebook containing all source code, final results, and final plots should be submitted via Canvas. To reiterate, **the graders should be able to see final results (plots, text, etc.) as soon as they open the notebook (i.e., before re-running the code).**

Carefully read the Course Syllabus for late submission and other policies. If you find yourself stuck on something, start a discussion on Canvas and/or come to TA office hours.