Link to weekly query submission: <https://docs.google.com/forms/d/e/1FAIpQLScyADTU7lkF2D5qYhp1vxOPVG6Rm2WS99iIAdQqRIkGft_2cQ/viewform?usp=preview>

Data and descriptions of functions for Bayesian optimisation competition

The time has finally come to begin the capstone project! A quick reminder that the project consists of a Bayesian Optimisation competition. There are eight unknown functions that you will be looking to optimise. For this project, all eight of the functions should be maximized. You will only be able to query each function once every few days, so you have to make every single opportunity count! The inputs for each query are the following:

* + Function one: 2-dimensional
  + Function two: 2-dimensional
  + Function three: 3-dimensional
  + Function four: 4-dimensional
  + Function five: 4-dimensional
  + Function six: 5-dimensional
  + Function seven: 6-dimensional
  + Function eight: 8-dimensional

In the [folder](https://classroom.emeritus.org/courses/10314/files/2856649?wrap=1)[Download folder](https://classroom.emeritus.org/courses/10314/files/2856649/download?download_frd=1)provided you will find initial data to guide your search. Please note that each function is synthetic (using real black-box functions would be too expensive for the contest to actually work). In the folder for each function, you will find some initial data to help you begin to search for the optima. The data is in NumPy format, so you can read it using np.load().

The descriptions provided below are to inform you about the properties of each function, and identify a few examples of where Bayesian Optimisation is applied in the real world. This background on each function also includes some hints towards finding a good solution.

*Select each tab to review.*

**Function 1: Searching for Contamination Sources**

This may sound simple because you only have a two-dimensional input, however it is a very difficult problem. It corresponds to trying to find the source of radiation in some square area. However, you can only detect the radiation once you are very close to it, meaning most of the readings will be zero. There are two sources, one is not too dangerous, so make sure you try to find both modes of the function.

**Function 2: Optimising Noisy Models**

This corresponds to trying to optimise an unknown machine learning model. However, the initialization of the model is very important, meaning your observations will be very noisy, and the problem might have a lot of local optima! You are trying to make the model’s log-likelihood as large as possible.

**Function 3: Drug Discovery Problem**

In this example, you are doing drug discovery! You can select three compounds to create a drug, and receive a measurement of the people’s adverse reaction to the drug. You want to make this as close as possible to zero. (hint: one of the variables may not cause any effects on the person).

**Function 4: Fast, but Inaccurate Modelling**

This example is for a particular business relying heavily on online sales. It can run very accurate calculations to figure out what is the optimal placement of their product across warehouses. Unfortunately, the calculations are extremely expensive (computationally) to run, so they can only do it once every two weeks. Instead, they propose using a machine learning model which approximates the solution quickly (in a few hours). The model has four hyper-parameters you need to tune, and the output corresponds to the difference between the expensive calculation, and the model. Since you are modelling a dynamical system, expect a lot of local optima!

**Function 5: Yield in a Chemical Reaction**

This time you are trying to optimise another four-dimensional black-box. It corresponds to the yield of a chemical process after processing in some factory. This type of process tends to be unimodal. Try to find the combination of chemicals that maximizes the yield!

**Function 6: Cake and Stuff**

Time to get cooking! You are optimising a cake recipe. There are five ingredients. The outputs correspond to the sum of different objectives: flavor, consistency, calories, waste and cost. Each objective receives negative points by our expert taster. You want this sum to be as close to zero as possible!

**Function 7: Sometimes Lazy is Best**

You are now optimising six hyper-parameters of a machine learning model. Note that it is a popular and frequently used model, so maybe you could search to see if anyone else has optisized it before?

**Function 8: High-dimensional Optimisation**

You’ve reach the final, 8-dimensional search space. High-dimensional black-box optimisation can be very difficult, so sticking to local solutions is not the worst idea here.