# 1 Machine Learning Algorithm and its implementation

Optimization algorithms commonly used are Gaussian process and Genetic Algorithm. Both of them are capable of optimizing the output of a unkown function when the evaluation cost, the running time of the function, is high.

Gaussian process ultilizes Bayesien Process and assumes each set of data we collect is normally distributed. It will suggest where the next data point to evaluate and will eventually lead to the global maximum in an ideal situation.

Genetic algorithm simulates what happens in real world. It considers each set of parameters as an "individual", and simulates the mating and mutation of the individual to select the best individual possibles. In this way, the "population", a group of individuals, will evolve and provide the best solutions to the optimization problem.

In this code, both method has been implemented. Gaussian process is implemented using GpyOpt library, and the Genetic Algorithm is with the help of "deap" library.

## 1.1 Optimization Goal

Here the optimization goal is to maximize the capture ratio of the 2D plus MOT, ignoring all other parameters, such as average velocity of the captured atoms. The parameters being optimized involve the height of the cooling beam with respect to the pushing beam, the detuning and e-radius of the cooling/pushing beams and the gradient of the magnetic field.

## 1.2 Implementation of the machine learning combined two algorithms

After some tries, I found out that the Gaussian process is not producing a lot of meaningful results. While some of the results are impressive and better than what I manually chose, many of the setups suggested by gaussian process produce zero capture ratio. A typical diagram of the evolution is given in the following graph.

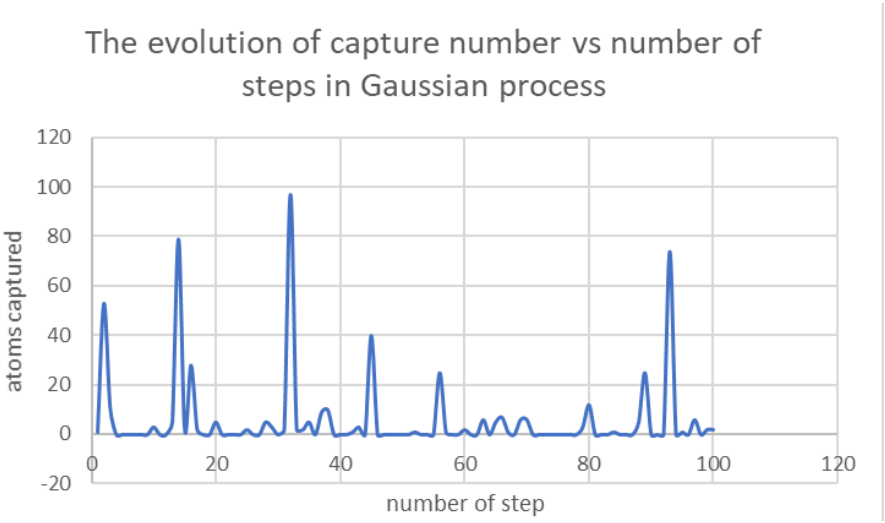


Figure 1: number of captured atoms as gaussian process progesses