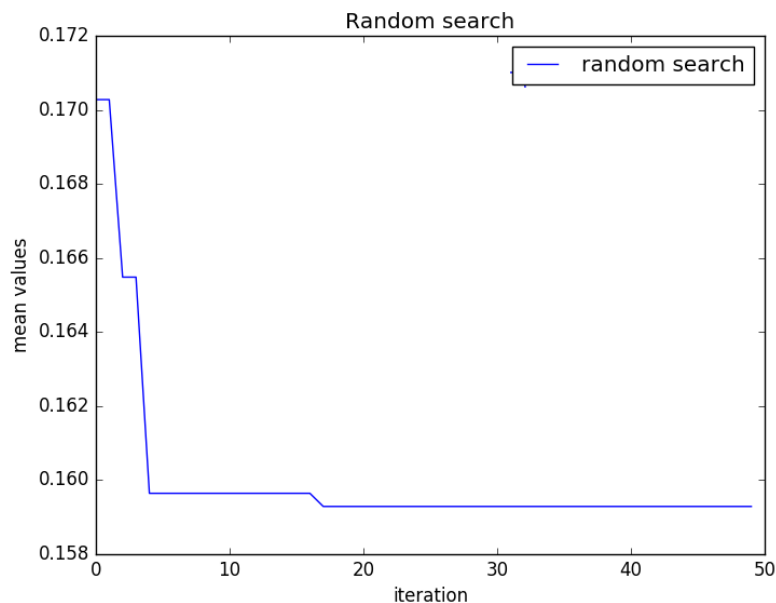


# Report for Assignment 5

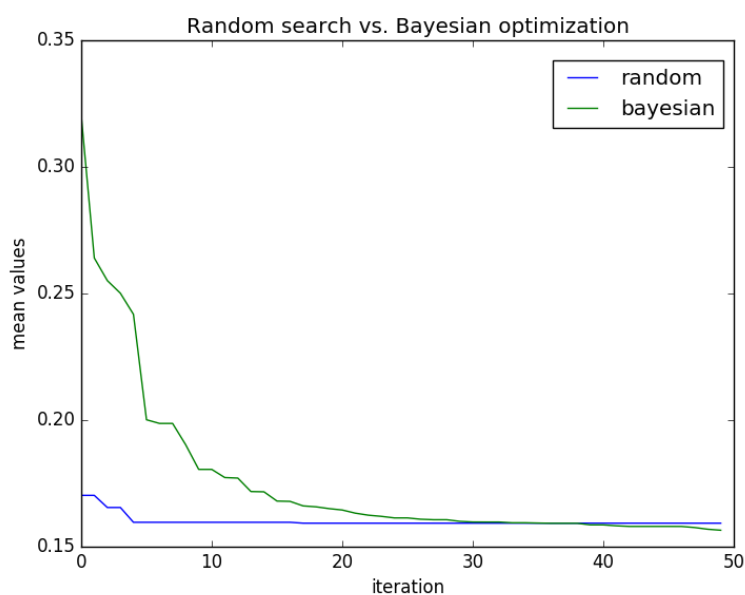
I implemented three methods: one for random search, one for Bayesian optimization and one for plotting. In the initialization of the class, I set the bounds for the hyperparameters. The most complex part was actually installing RoBo and tuning the method to work with the framework.

The plot for the random search is as follows:



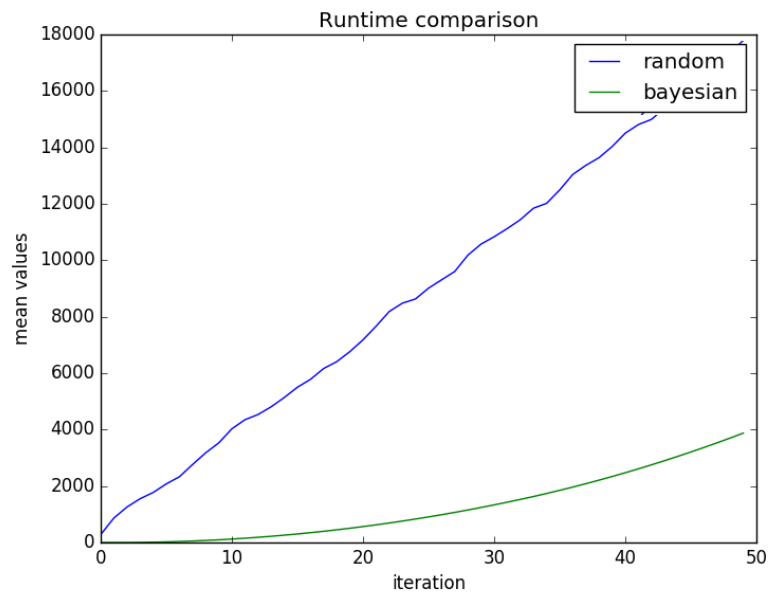
Most of the runs for random search look like that. However, given the arbitrary nature of the method, it could have found the best value in the beginning. The graph would then be a line parallel to the x-axis.

The performance of both methods plotted against each other:



Here we can see how initially Random search performs better (until the 20<sup>th</sup> – 30<sup>th</sup> iteration), but in the end Bayesian optimization finds a better result. This is so because Bayesian follows a more gradual approach, while random search usually gets lucky in the first iterations.

The cumulative runtime for both methods:



Random follows a more linear runtime, but it is much worse than Bayesian. The actual runtime is still smaller than what it would have taken a proper network, thanks to the surrogates.