Deep Learning Lab 2018/19

Exercise 4 : Automated Machine Learning

Name: Jatin Dhawan

Matriculation No. : 452322

Introduction: The aim of this exercise was to learn the implementation of both Bayesian optimization and Hyperband. And finally how to combine both to perform the hyperparameter optimization of a CNN on CIFAR-10 dataset. We were given a surrogate (random forest as regression model) benchmark to avoid the training up of a CNN network and to evaluate the hyperparameter configuration faster. The figure 1 illustrates the results vs time of different hypermater optimization methods. BOHB not only be the fastest amongst all, but also converges to the optimum value.

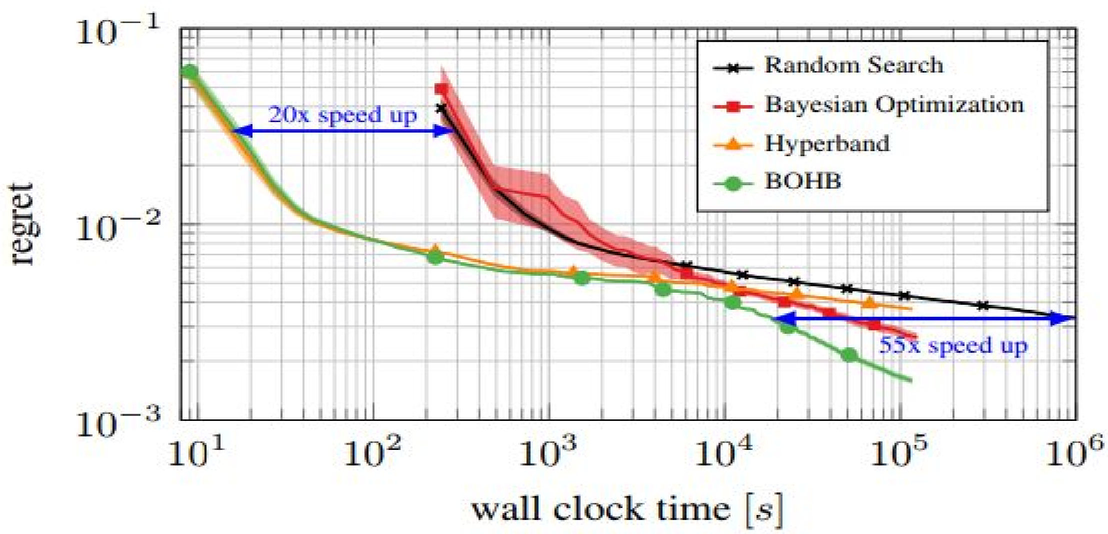
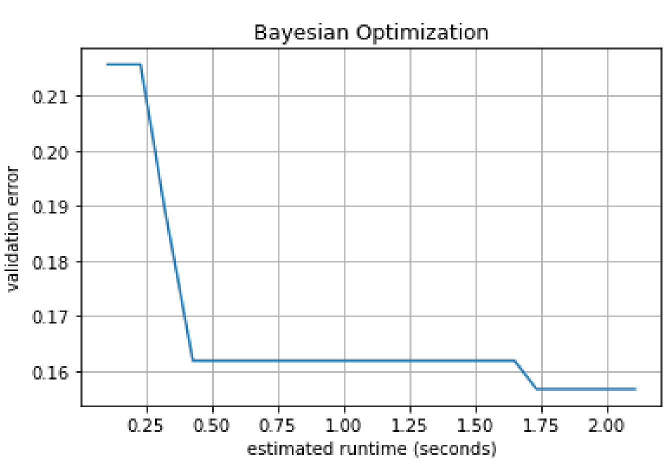


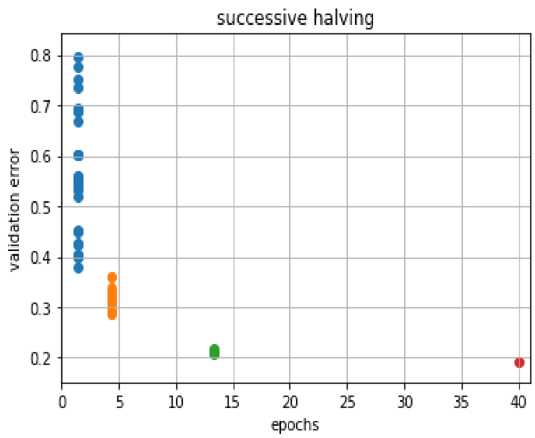
Figure 1: Automating the ML pipeline

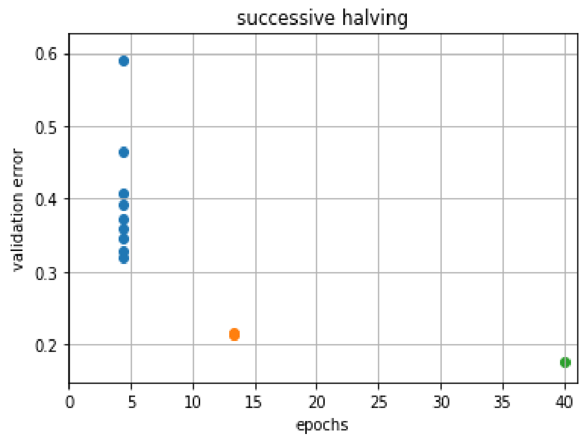
Bayesian optimization: It internally uses a model to guide the search. In our code, we randomly generated some hyperparameter configurations. BO uses Gaussian processes(GP) as a objective function that evaluates our validation error, Expected Improvement(EI) as an acquisition function which selects the next set of hyperparameters and an optimizer to optimize our acquisition function. After performing the figure 2 shows the runtime trajectory and the validation error after 20 iterations.

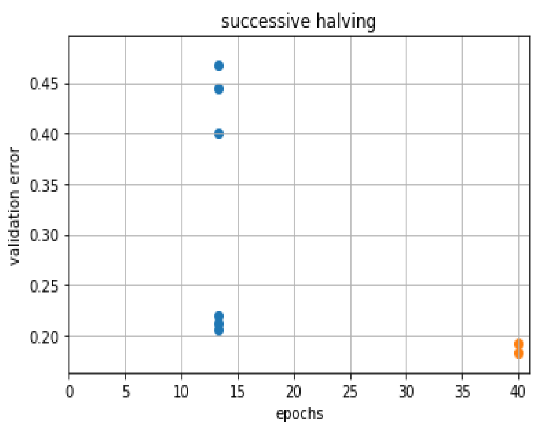


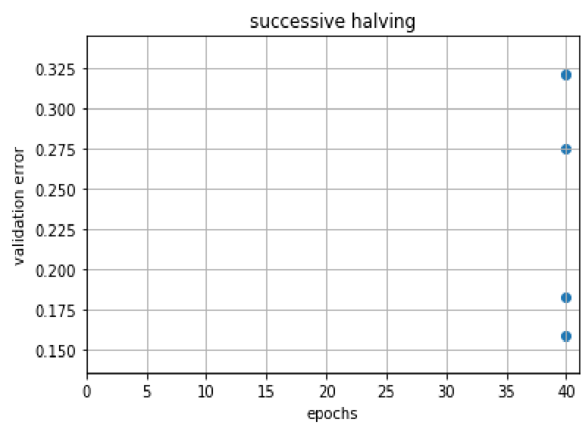
Global Optimum : 0.156685734563

Hyperband: In order to speed up the optimization process , hyperband uses different fidelities. We used our learning curve as a fidelity. Hyperband doesn't rely on previous model configuration as bayesian but samples configurations randomly. Hyperbands combines random search with successive halving to balance very aggressive evaluation with many configurations on the smallest budget and very conservative runs on the maximum budget.

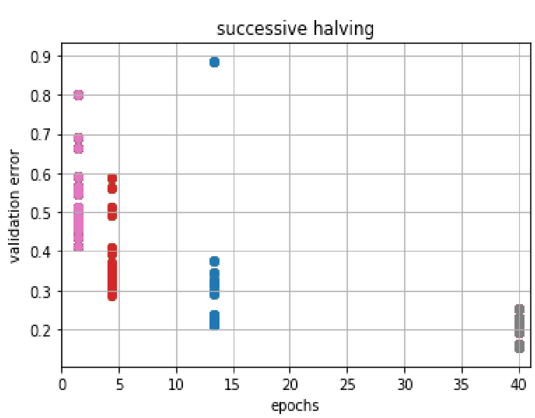


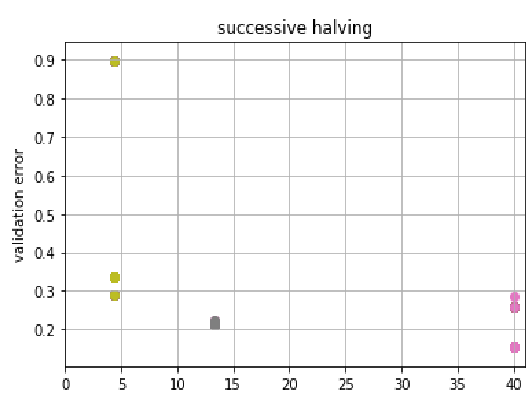


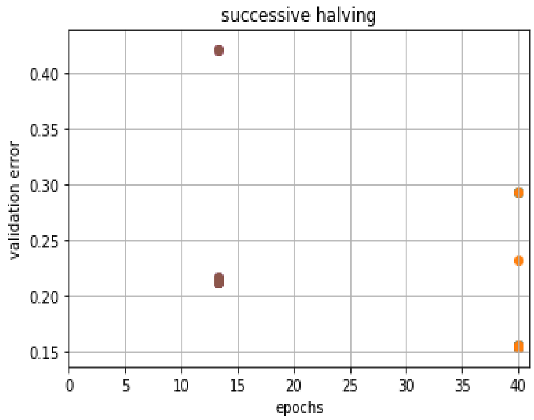


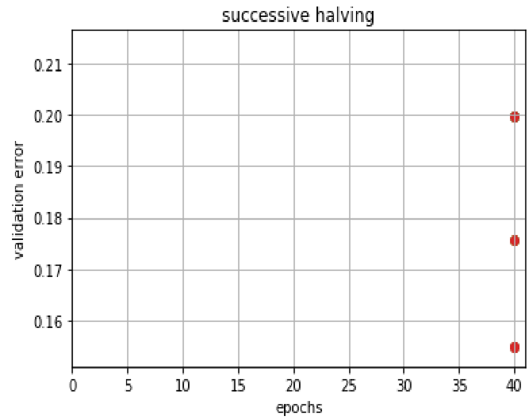


BOHB: One of the weaknesses of Hyperband is that it draws configurations randomly and could take exponentially long to approach the global optimum. We combined Hyperband with a kernel density estimator that models the distribution of the good and the bad configurations in the input space. By sampling from this model instead of a uniform distribution we can find good configurations much faster. Figure below shows the successive halving with the incumbent trajectory for each configuration whether it was sampled randomly or from our model.









Conclusion: After evaluating the random with model based BOHB, we can see that BOHB not only converges to minimum fast but also achieves global optimum.