Submission for Deep Learning Exercise

Team: shallow_learning_group
Students: Batuhan Karaca

January 28, 2024

Black-box Optimization with Random Search

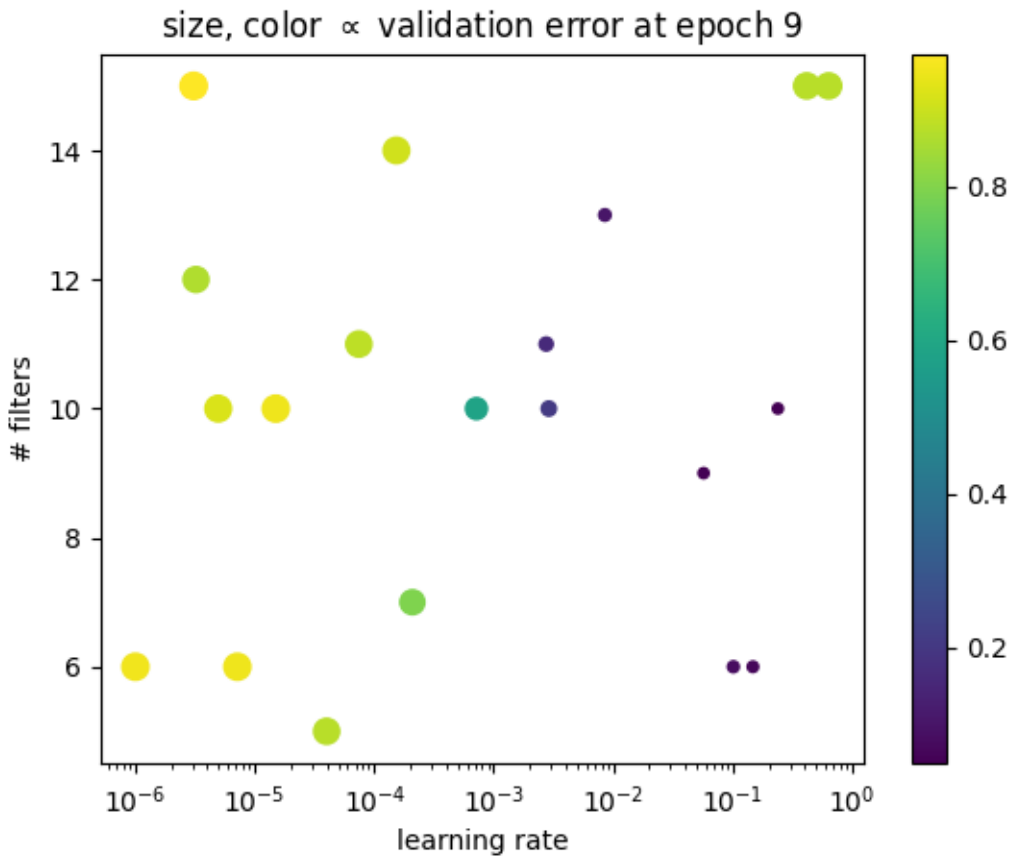

Figure 1: Scatter plot of validation error for different combination of hyperparameters for random search algorithm, at epoch 9

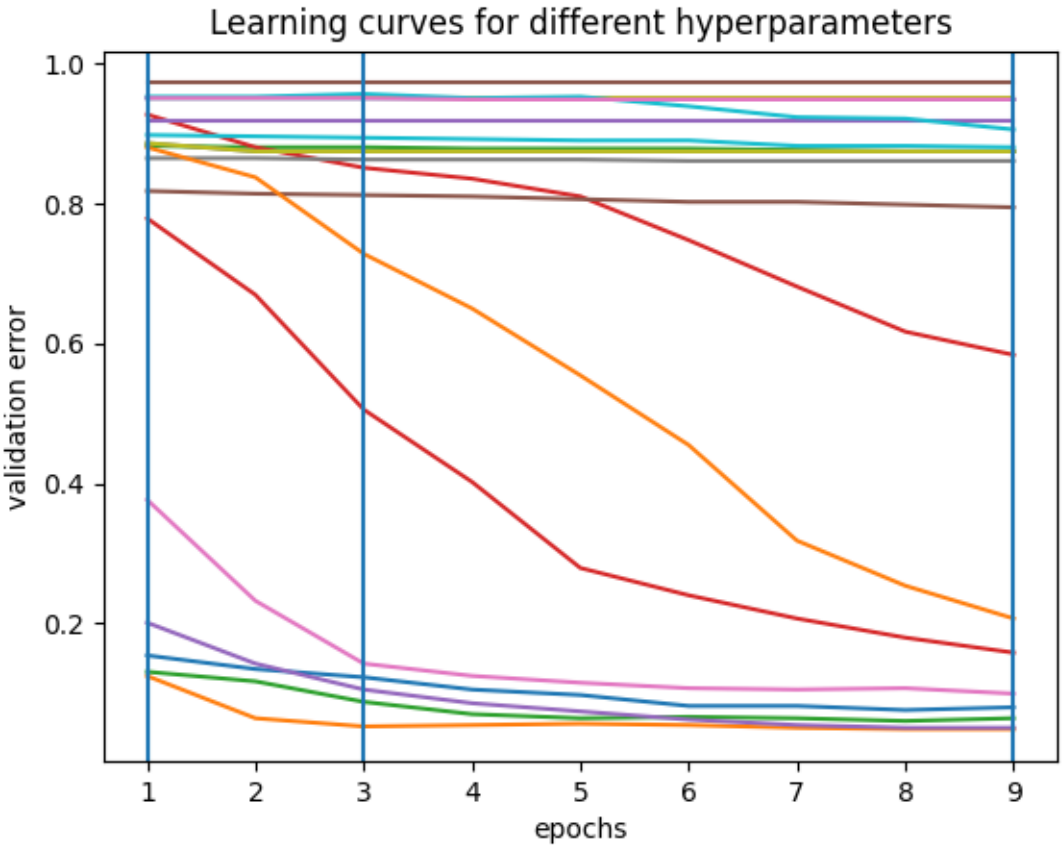

Figure 2: Line plot of validation error for different hyperparameters for random search algorithm

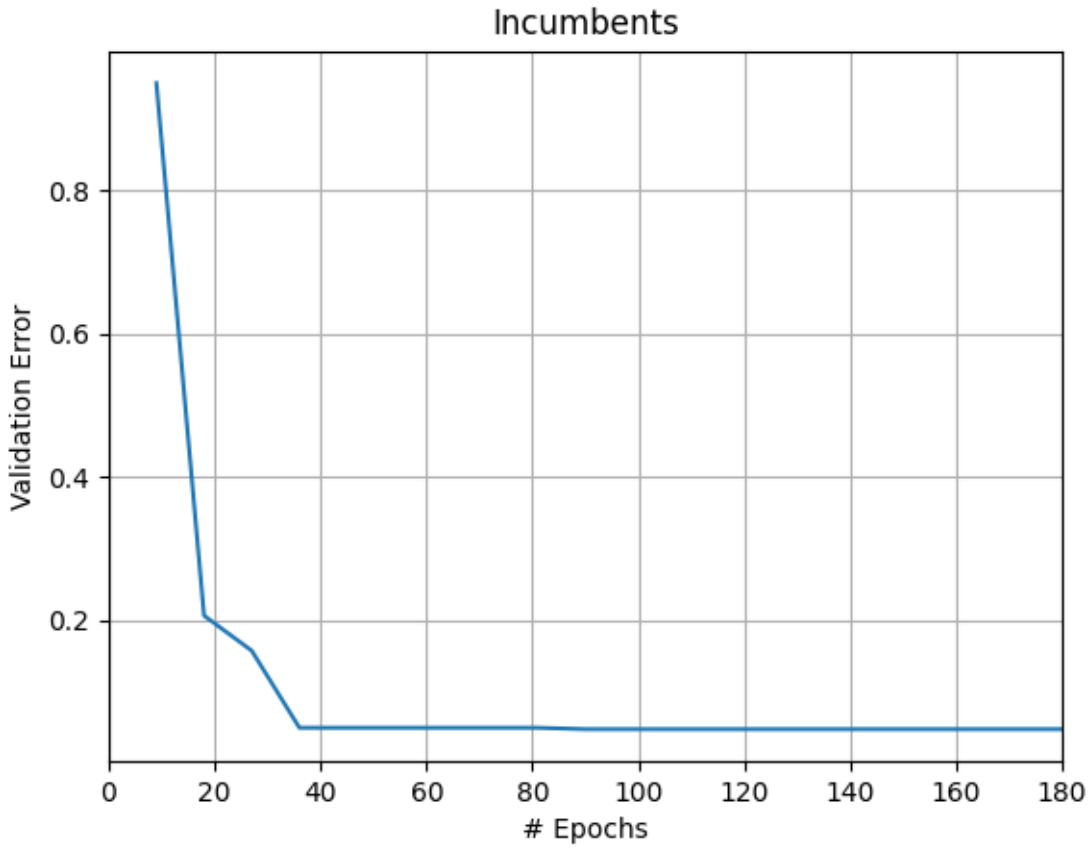

Figure 3: Line plot of validation error for incumbents for random search algorithm

What pattern do you see for the scatter plot? Why might it occur?

Scatter plot in figure 1 shows a relationship between combinations of (Number of filters, Learning rate) hyperparameter pairs versus validation error at epoch 9. For each task, generally there is an interval in which learning rate is neither too small nor too large that the network performs the best. For the smaller values, the convergence rate is so slow that more training epochs are required. For this graph we cannot see; however, for larger values of learning rate, the loss fluctuates around the true gradient with larger diameter, potentially ending up in local (non-optimal) minima. When number of filters increases, number of parameters becomes larger increasing the capacity, potentially overfitting the model. When model both overfits and finds non-optimal minima, it usually performs worse, as shown in the scatter plot.

Given the error curves would you expect multi-fidelity optimization or black-box optimization to perform better?

Because loss spaces of different fidelities are generally similar and multi-fidelity algorithms operate more on cheaper fidelities, they should find better performing hyper-parameters in shorter amount of time. Looking at figures 2, 4 and 6 we see that the HyperBand and PriorBand find combinations that perform better in same amount of time. Furthermore, their incumbents in figures 5 and 7 start converging sooner as opposed to random search in figure 3, supporting the claim.

Multi-fidelity Optimization with HyperBand

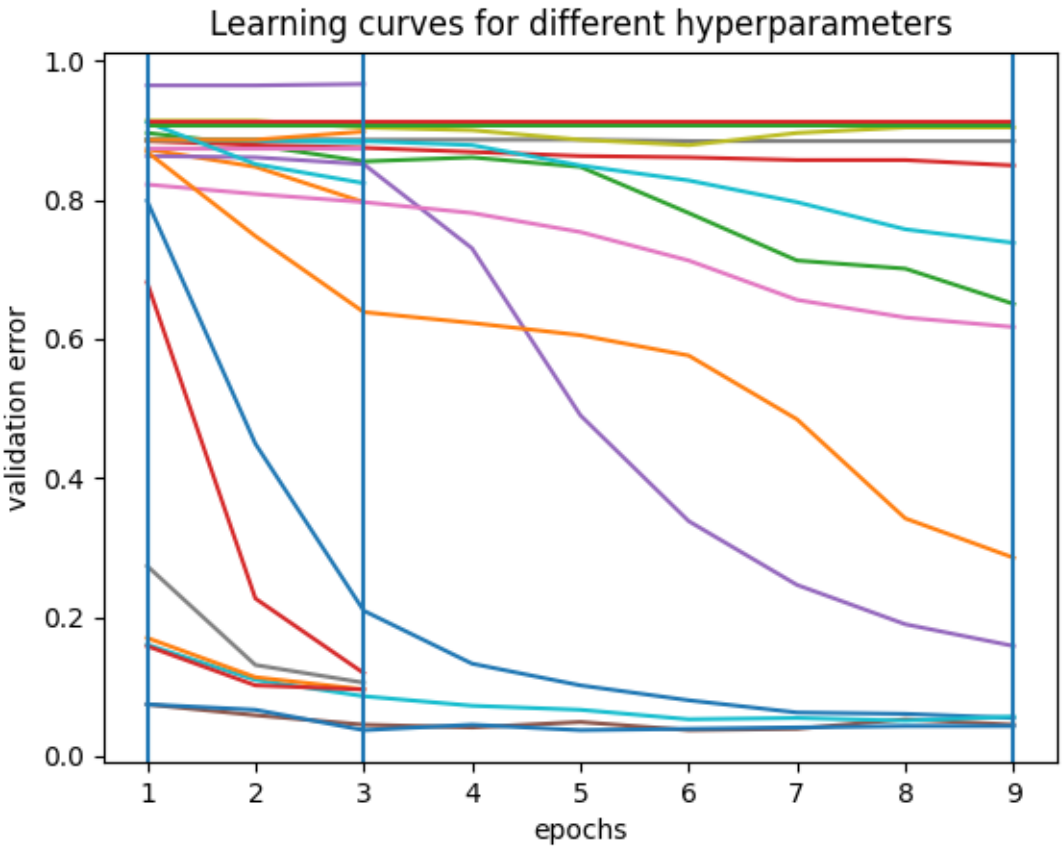

Figure 4: Line plot of validation error for different hyperparameters for HyperBand algorithm

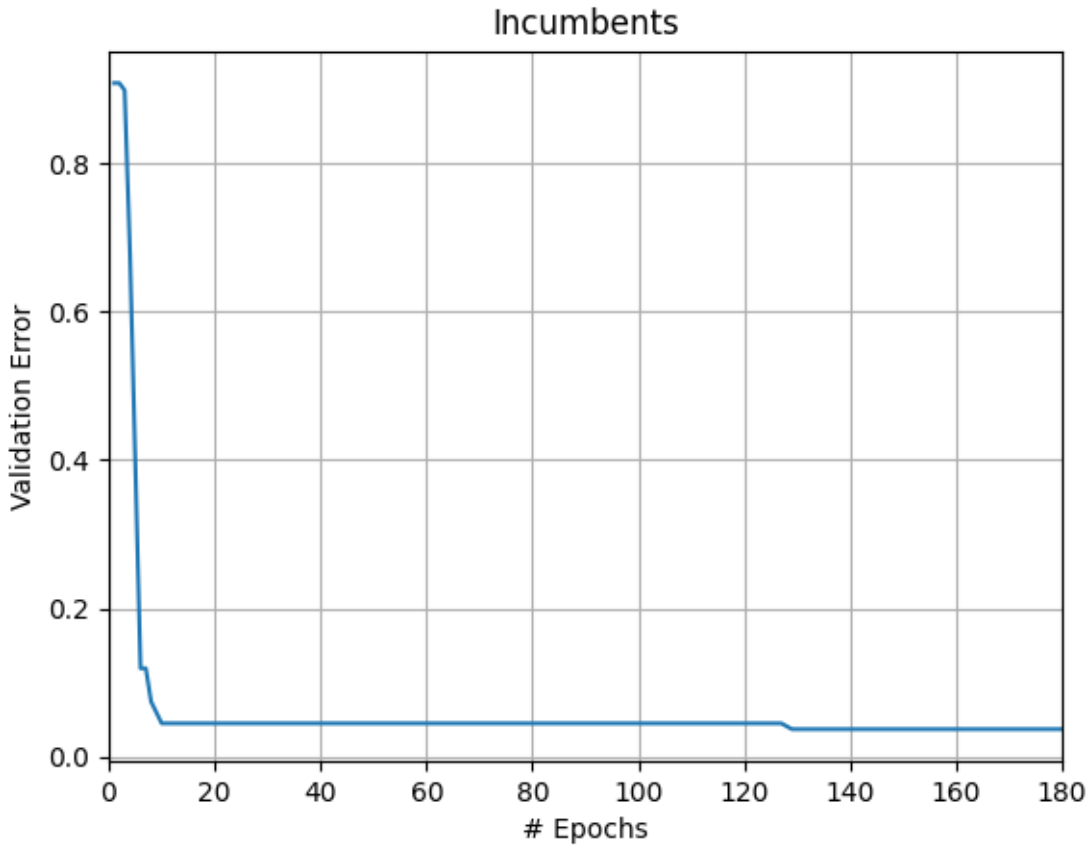

Figure 5: Line plot of validation error for incumbents for HyperBand algorithm

Did HyperBand always early-stop the configurations with the worst final performance? Why?

At each iteration, HyperBand selects the configurations randomly. Therefore, there could be some poor configurations selected as well.

Multi-fidelity Optimization with PriorBand

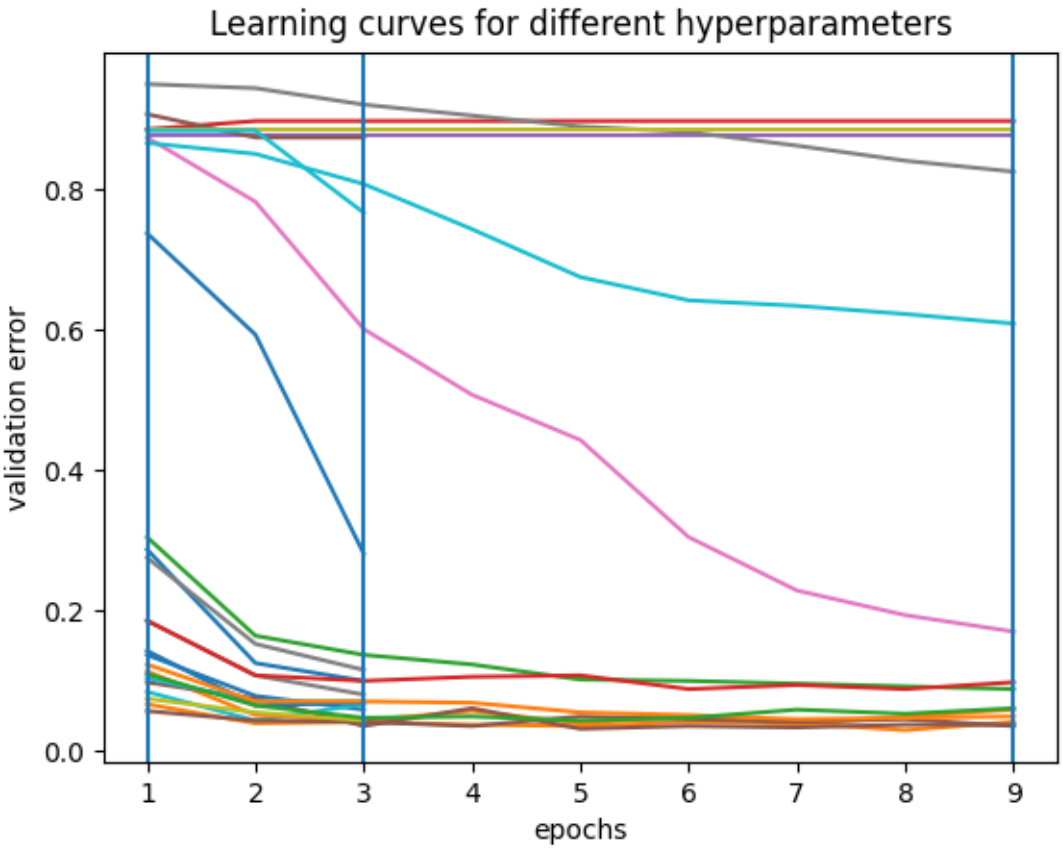

Figure 6: Line plot of validation error for different hyperparameters for PriorBand algorithm

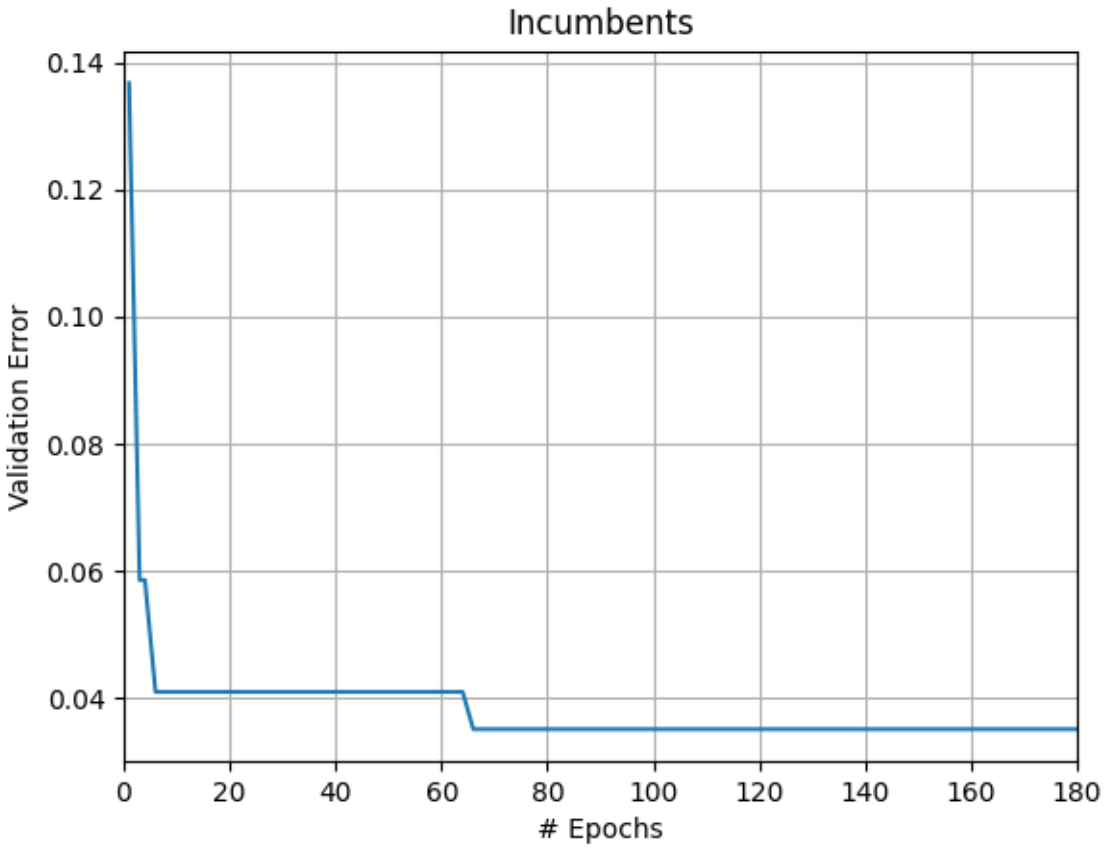

Figure 7: Line plot of validation error for incumbents for PriorBand algorithm

Did PriorBand always early-stop the configurations with the worst final performance? Why

Even though PriorBand samples its configurations using its priors unlike HyperBand, the priors may not be optimal at all.

Comparison of Random Search, Hyperband, and Priorband

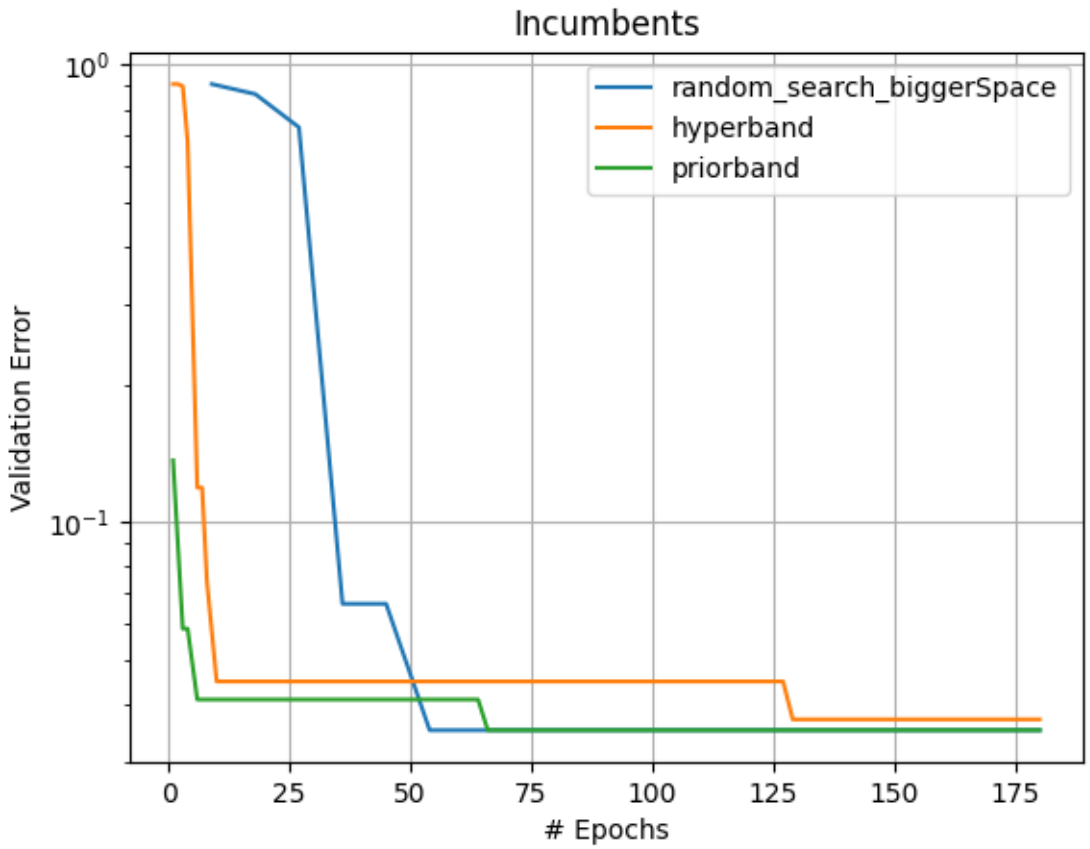

Figure 8: Line plot of validation error for incumbents of random search, HyperBand and PriorBand algorithms for comparison

Give an interpretation of the plot.

Looking at figure 8, we see PriorBand performed the best and random search performed the worst. Both multi-fidelity algorithms start converging sooner with a larger margin compared to the random search. At the long run, all algorithms have more or less the similar values as random search in average will sample every possible value in a discrete finite region (bounded by some upper and lower values). Priorband did slightly better compared to vanilla HyperBand.

In this case which of the optimizers was more efficient and why?

As mentioned, due to the fact that loss spaces of different fidelities are generally similar and multi-fidelity algorithms operate more on cheaper fidelities, they should find better performing hyper-parameters in shorter amount of time. Furthermore, PriorBand uses its priors as opposed to sampling the configuration at random unlike HyperBand. Given that these priors are good, it finds minima more quickly than HyperBand.