# Joint Architecture Search and Hyperparameter Optimization of a CNN

**Chuhuan Shen**
Leibniz University Hannover

Leibniz Universität Hannover

## 1 TL;DR

**This project provides a performance improvement over the provided baseline using BOHB for the Joint Architecture and Hyperparameter Search.**

## 2 Motivation

- The performance of a neural network is influence by two factors:
  - Neural network architecture
  - Hyperparameters
- Neural architecture search (NAS) methods typically consider fixed hyperparameters, and hyperparameter optimization (HPO) methods typically consider a fixed neural architecture
- In this project, the neural architecture is considered a special hyperparameter and aims to use BOHB for Joint Neural Architecture Search and Hyperparameter Optimization(JAHS).

## 3 Search Space

- To be consistent with the provided baseline, the experiments were conducted using the same simple CNN architecture.

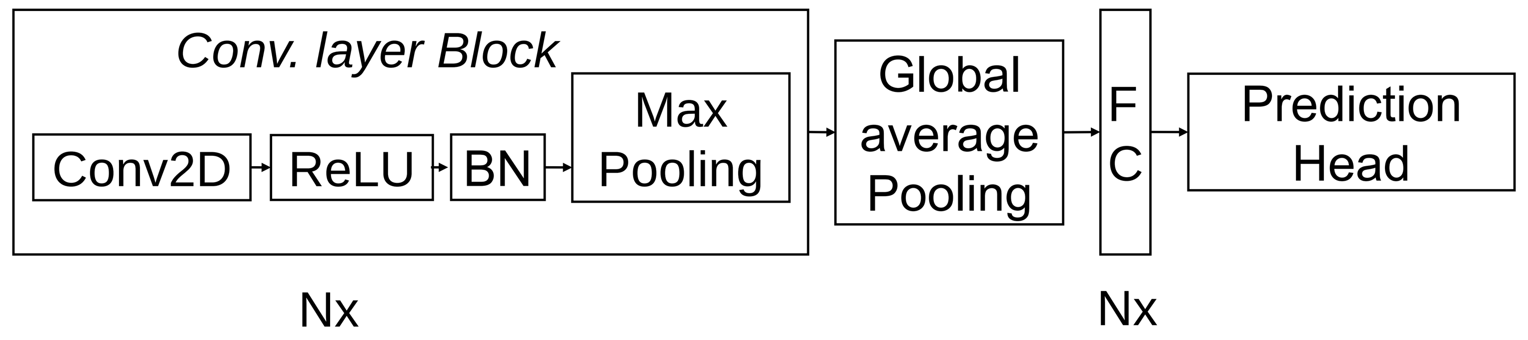



Figure1: Overview of the simple CNN architecture used for the experiments

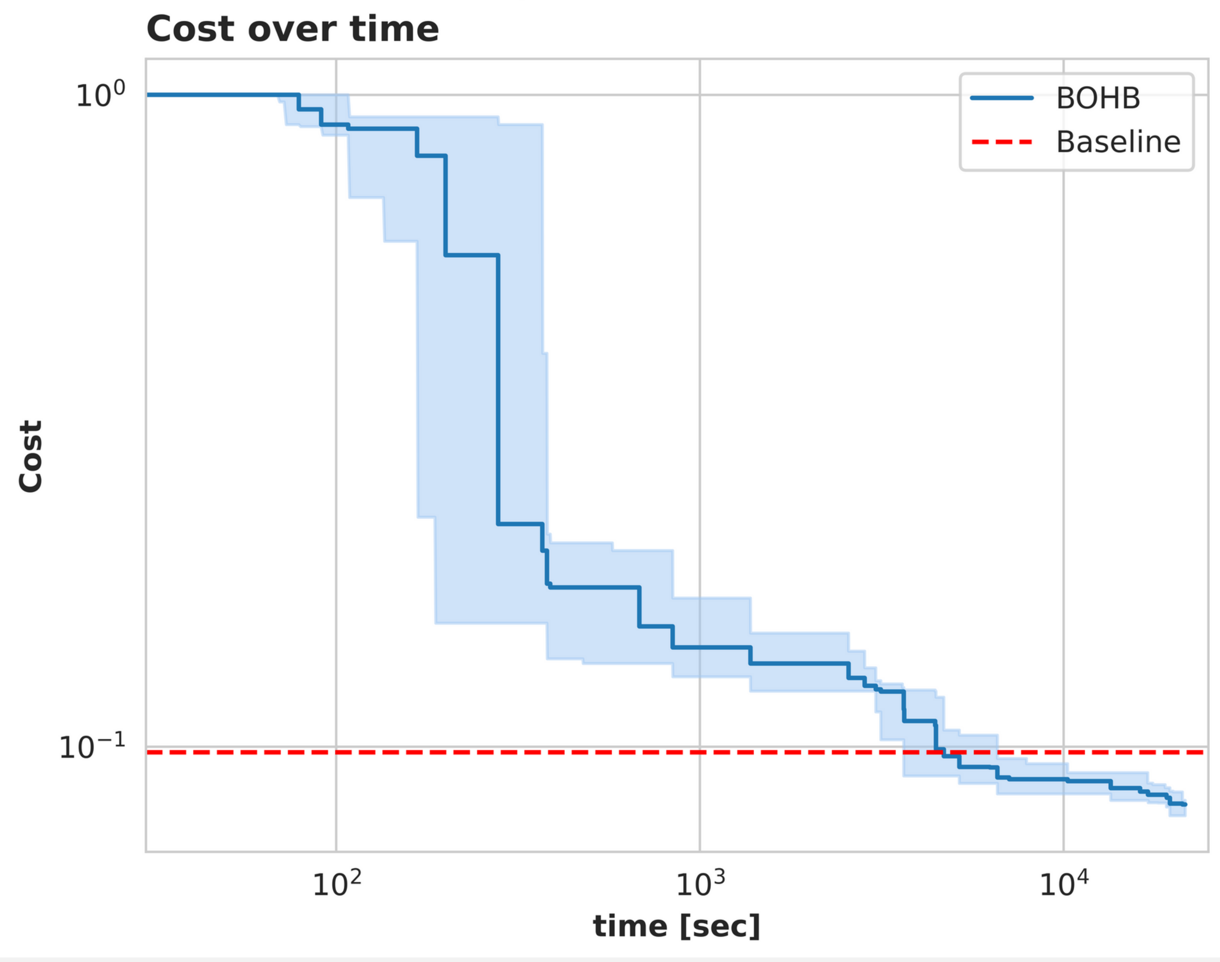
| Space | Description | Value property |
|---|---|---|
| Architecture | Number of convolutional layers | {1, 2, 3} |
| | Number of filters in a convolutional layer | [8, 256] |
| | Kernel size | {3, 5} |
| | Batch normalization | {True, False} |
| | Global average pooling | {True, False} |
| | Number of fully connected layers | {1, 2} |
| | Number of neurons in a fully connected layer | [512, 2048] |
| Hyperparameter | Batch size | {64, 128, 256, 512, 1024, 2048} |
| | Learning rate | $[10^{-3}, 10^{0}]$ |
| | Weight decay | $[10^{-5}, 10^{-2}]$ |
| | Dropout rate | {0.1, 0.2, 0.3, 0.4, 0.5} |
| | Optimizer | {Adam, SGD} |
| | Random horizontal flip | {True, False} |
| | Random rotation | {True, False} |

Table1: The joint search space of the architecture and hyperparameters

## 4 Experiments

The graph of performance over time shows that there is still potential for performance improvement with more time given.

Figure2: Cost over time distribution of 8 runs (median and 25=75%-quartiles range)



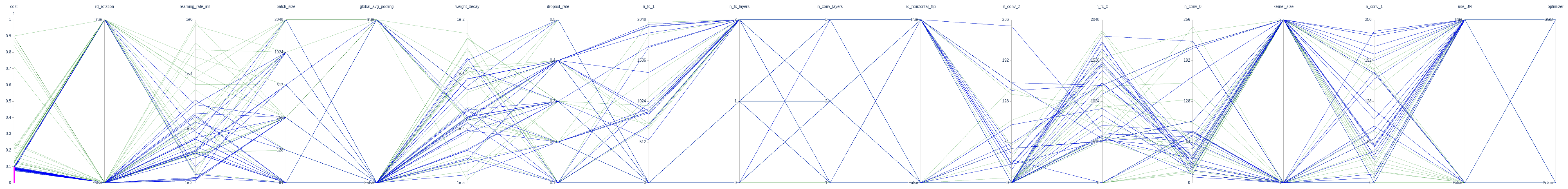

- No well-performing configuration uses global average pooling after using batch normalization.
- Well-performing configuration prefer using mid/small size mini-batch and filters/neurons.
- Interestingly, none of the final incumbents used Random rotation(30°) in the data augmentation phase.

All 8 runs beat the provided baseline accuracy by a certain margin. On average, there was a 1.8% improvement over baseline.

| Method | Test Accuracy (%) |
|---|---|
| Baseline | 90.5 |
| BOHB | 92.3 ± 0.4 |

Table2: Best Incumbent's test performance (50 Epochs with early stopping, average of 8 Random Seeds)

Figure3: Parallel Coordinate Plot of all evaluated configuration. (Coordinates are ordered from left to right according to hyperparameter importance. The blue line indicates all configurations with missclassication errors lower than the baseline. )



## 5 Conclusion & Future Works

- This project demonstrates that BOHB can steadily improve the performance of a simple model over the provided baseline in a limited amount of time
- Adding hyperparameter importance analysis to the workflow of AutoML to increases the interpretability.
- Extending the search space of neural architecture to form more sophisticated CNN architecture.
- Taking the number of model parameters into account turns JAHS into a multi-objective optimization task to find efficient and well-performing models.

github