Automated Machine Learning
Summer Semester 2021
Final Project

Carlos Marañes

Motivation

Problem statement: maximize a network accuracy with the following constraints:

- Network size (upper bound)
- Precision (lower bound)

These constraints are needed, e.g., in domains with **hardware restrictions** or **medical applications**, where precision is crucial

Motivation

Problem statement: maximize a network accuracy with the following constraints:

- Network size (upper bound)
- Precision (lower bound)

These constraints are needed, e.g., in domains with **hardware restrictions** or **medical applications**, where precision is crucial

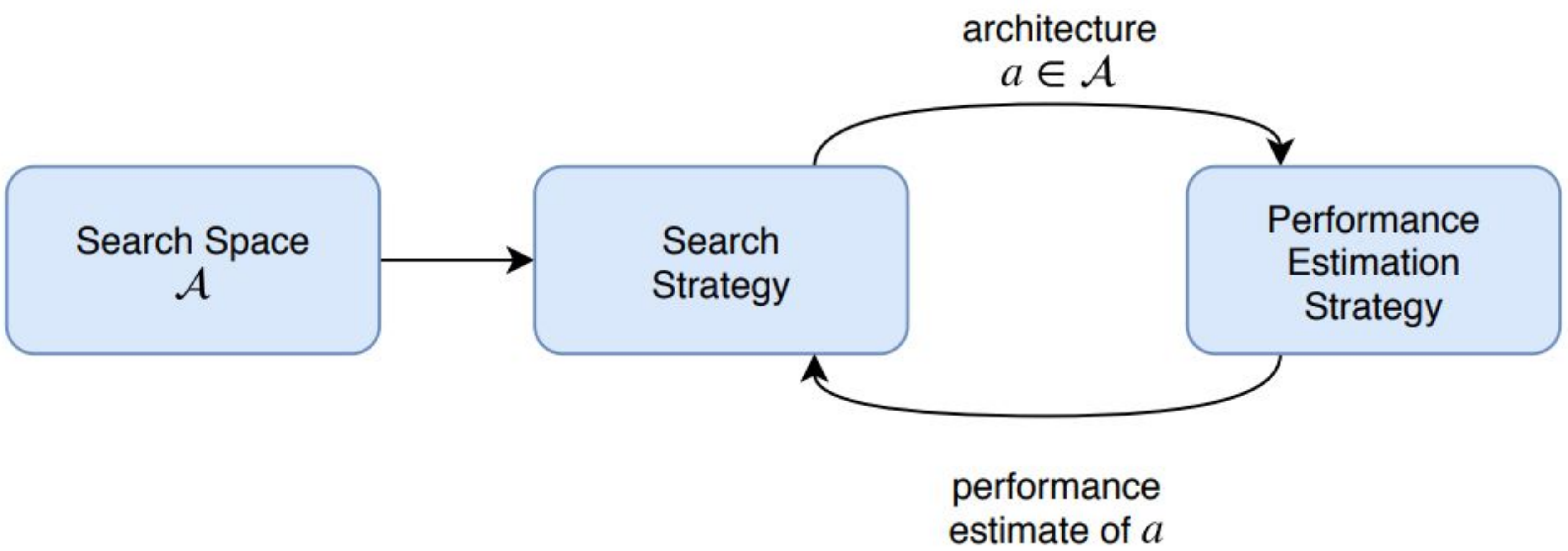
In this case, we are going to optimize a flower classifier

https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

Approach

Task is solved as a **NAS problem**. Needed of define:

1. Search Space
2. Search Strategy
3. Performance Estimation Strategy

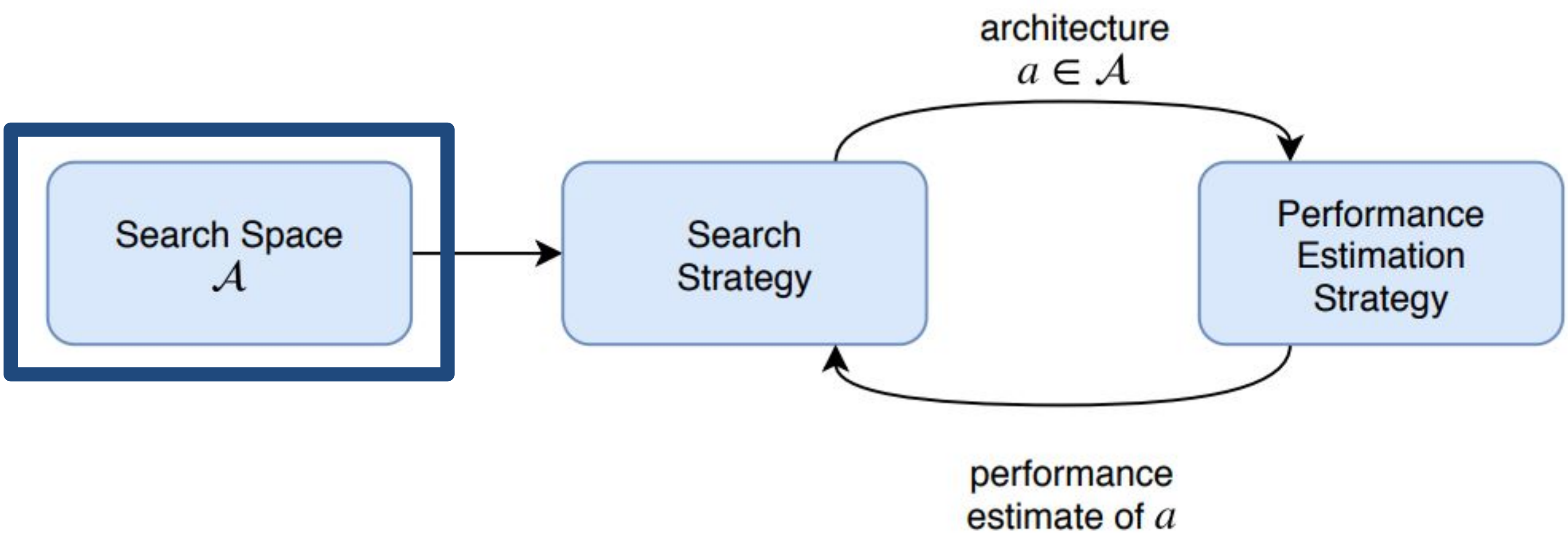


[AutoML Course]

Approach

Task is solved as a **NAS problem**:

1. **Search Space**
2. Search Strategy
3. Performance Estimation Strategy



[AutoML Course]

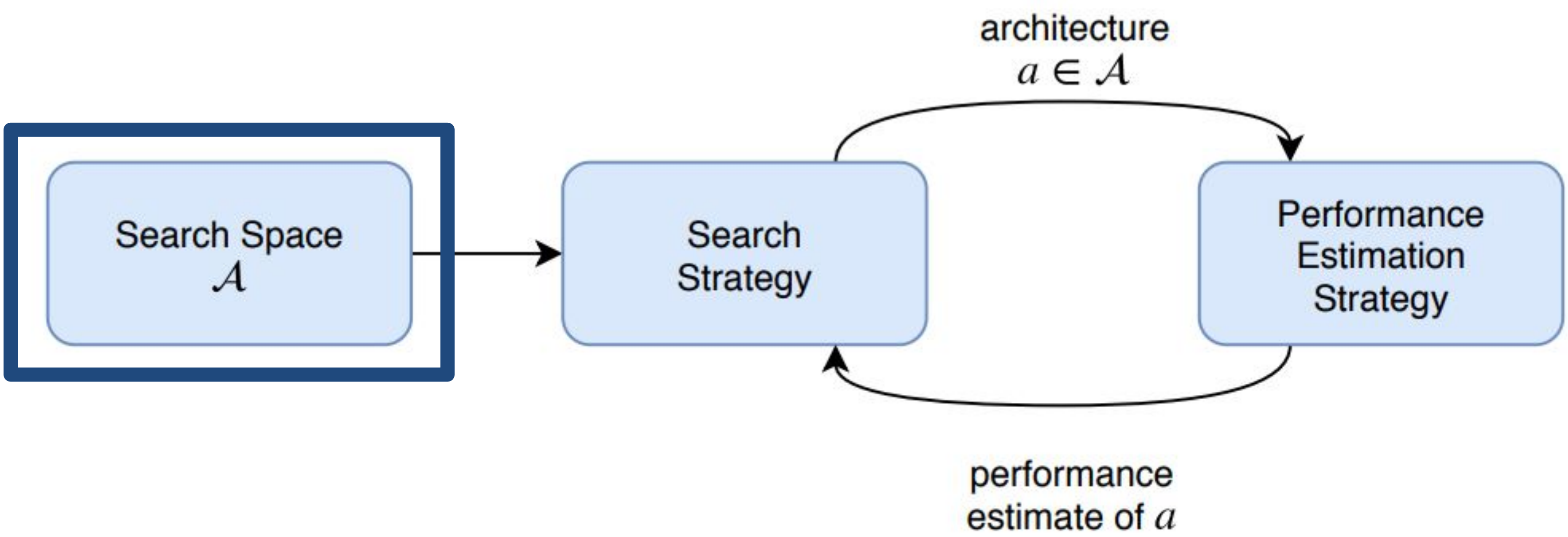
Approach - Search Space

Hyperparameter	Values	Log Scale
Learning rate	[0.00001, 0.1]	True
Batch size	[4, 300]	True
Kernel size	{3, 5, 7}	False
Batch Norm	{True, False}	False
Global average pooling	{True, False}	False
Dropout rate	[0.01, 0.5]	True
Number of convolutional layers	{1, 2, 3}	False
Number of channels of convolutional layers	[16, 1024]	True
Number of fully connected layers	{1, 2, 3}	False
Number of channels of fully convolutional layers	[4, 512]	False

Approach

Task is solved as a **NAS problem**:

1. **Search Space**
2. Search Strategy
3. Performance Estimation Strategy

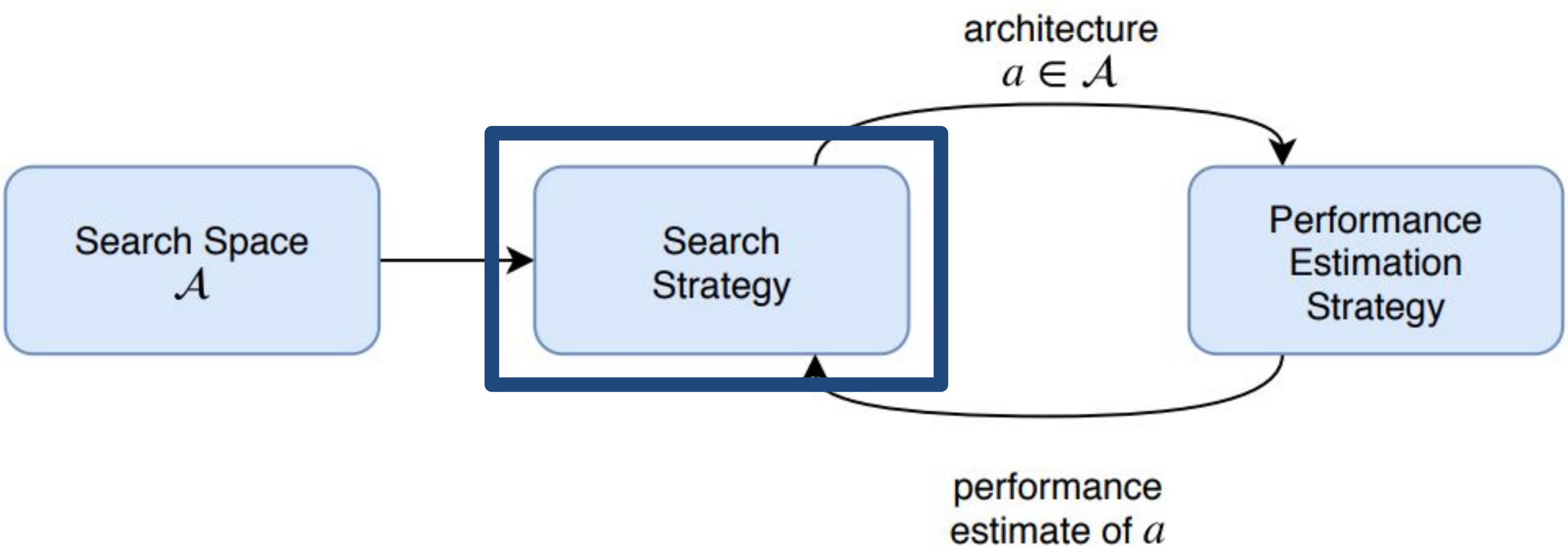


[AutoML Course]

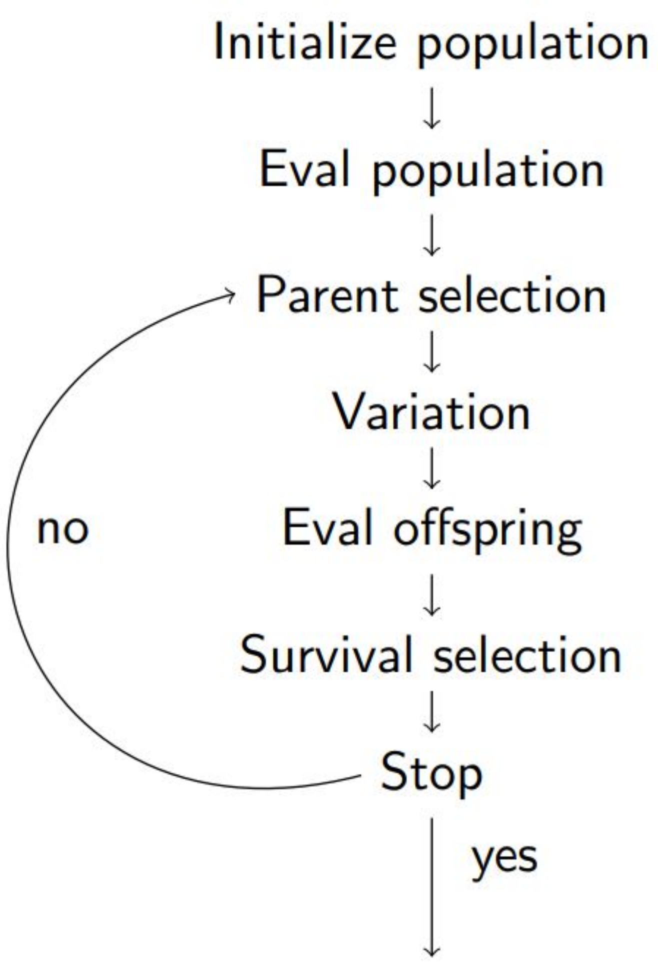
Approach

Task is solved as a **NAS problem**:

1. Search Space
2. **Search Strategy**
3. Performance Estimation Strategy



[AutoML Course]



[AutoML Course]

Evolutionary Algorithms

- Simple and powerful [Awad et al. 2021]
- Some changes to make it work with constraints

Approach - Search Strategy

1. Idea: Prioritize a population that satisfy the constraints and then improve accuracy
2. Models that do not satisfy the maximum number of parameters will be discarded
   - **A-priori sorting**. First precision and then accuracy

Approach - Search Strategy

1. Idea: Prioritize a population that satisfy the constraints and then improve accuracy
2. Models that do not satisfy the maximum number of parameters will be discarded
   - **A-priori sorting**. First precision and then accuracy

Individual

[Accuracy, Precision] (min precision: 0.39)

Approach - Search Strategy

1. Idea: Prioritize a population that satisfy the constraints and then improve accuracy
2. Models that do not satisfy the maximum number of parameters will be discarded.
   - **A-priori sorting**. First precision and then accuracy

Individual

[Accuracy, Precision]

(min precision: 0.39)

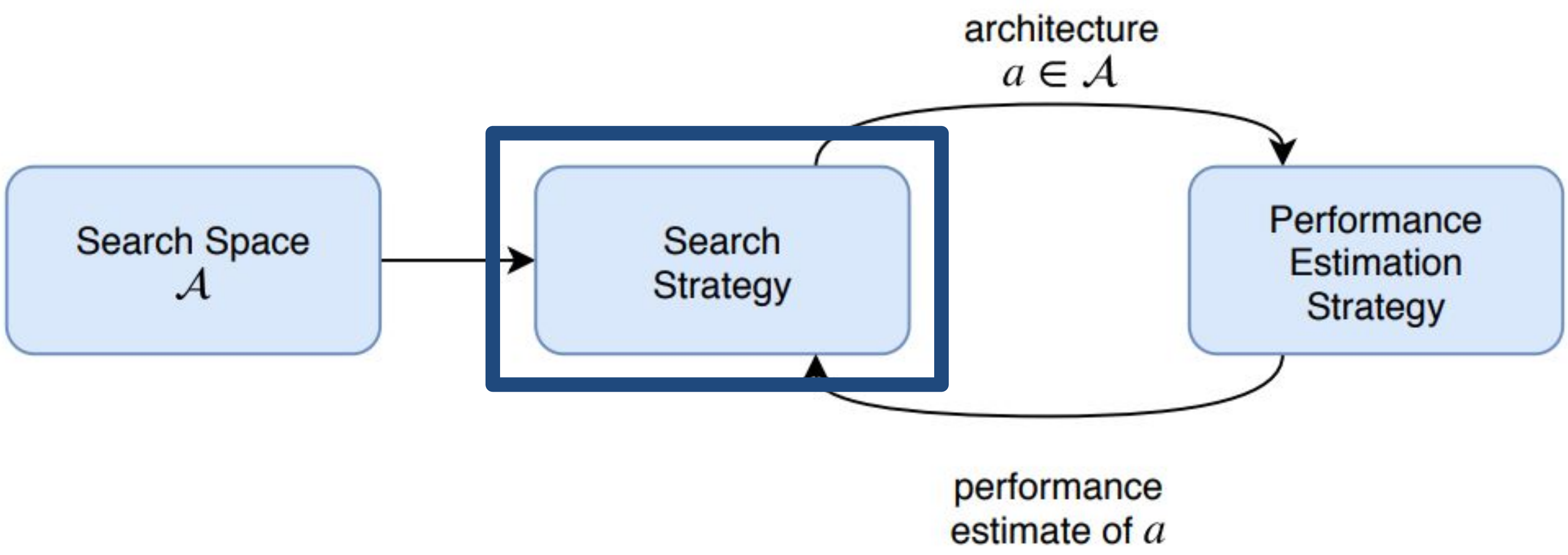
[0.85, 0.45] > [0.80, 0.46] > [0.75, 0.3] > [0.95, 0.2]

Population

Approach

Task is solved as a **NAS problem**:

1. Search Space
2. **Search Strategy**
3. Performance Estimation Strategy

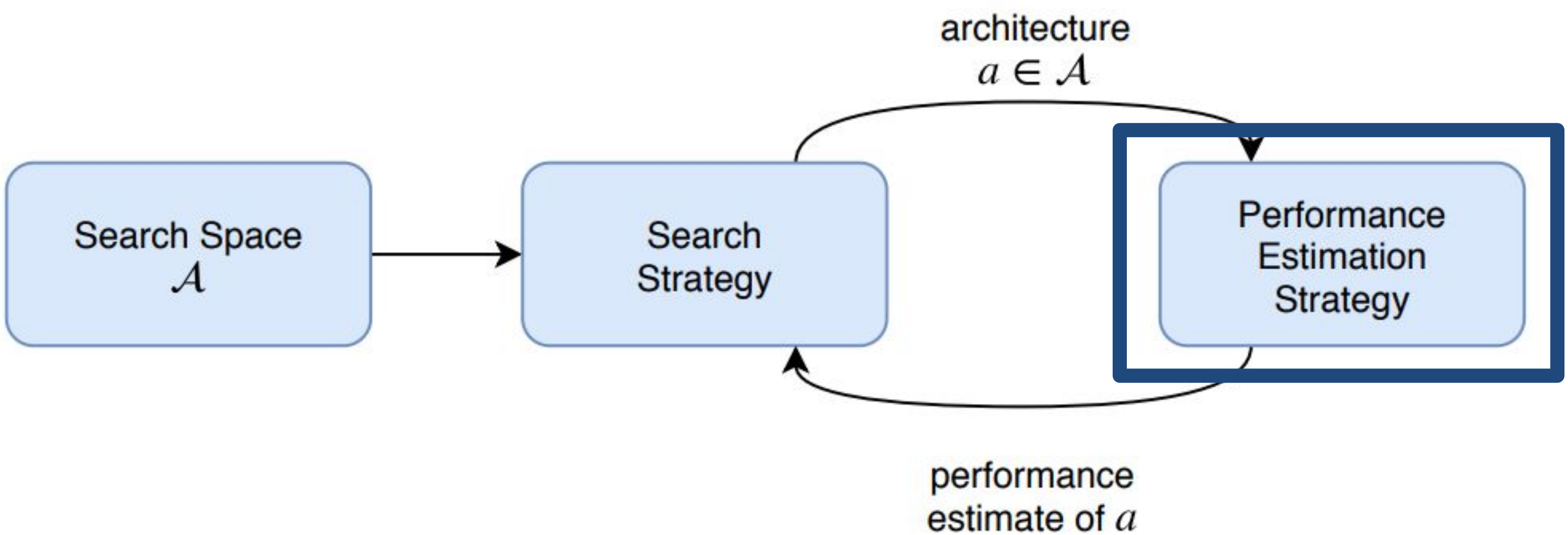


[AutoML Course]

Approach

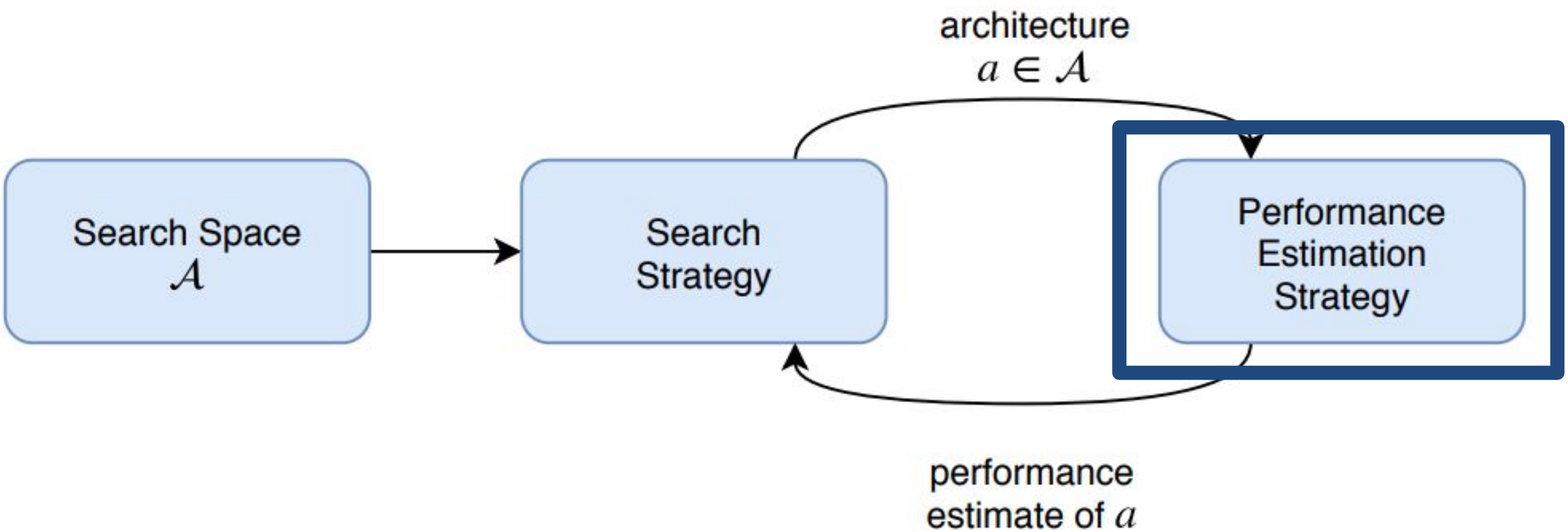
Task is solved as a **NAS problem**:

1. Search Space
2. Search Strategy
3. **Performance Estimation Strategy**



[AutoML Course]

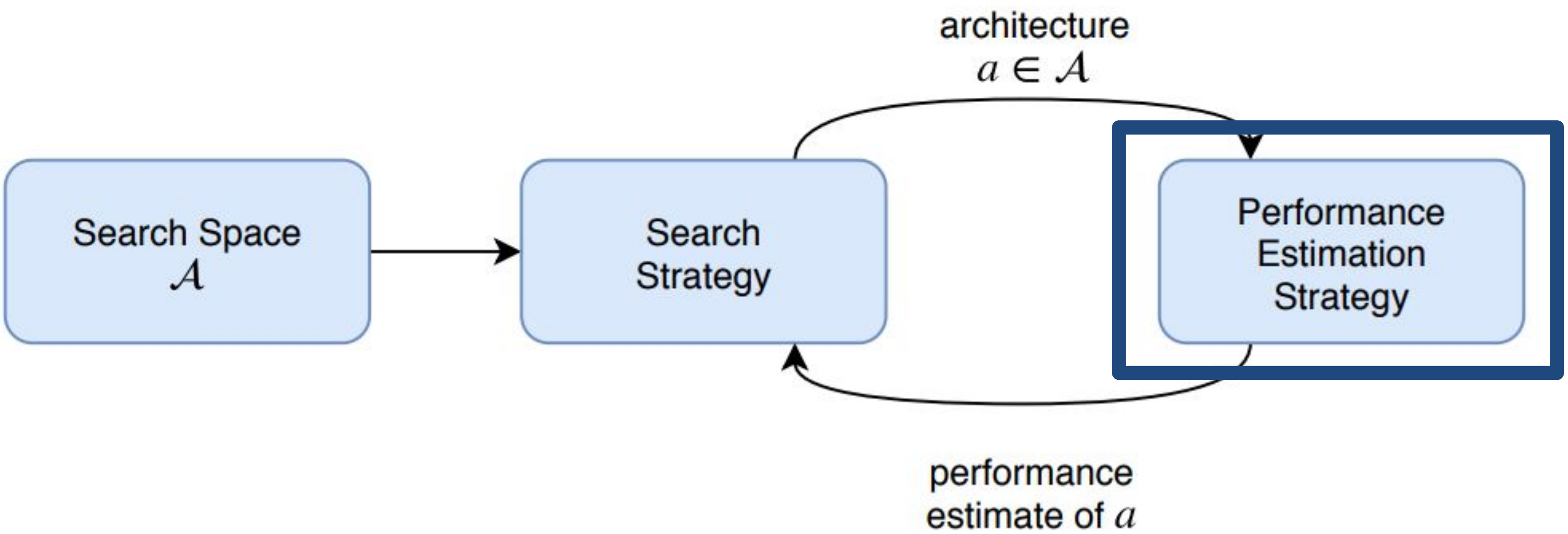
Approach



With the current approach we are fully evaluating individuals that may not satisfy the precision constraint…

[AutoML Course]

Approach



With the current approach we are fully evaluating individuals that may not satisfy the precision constraint…

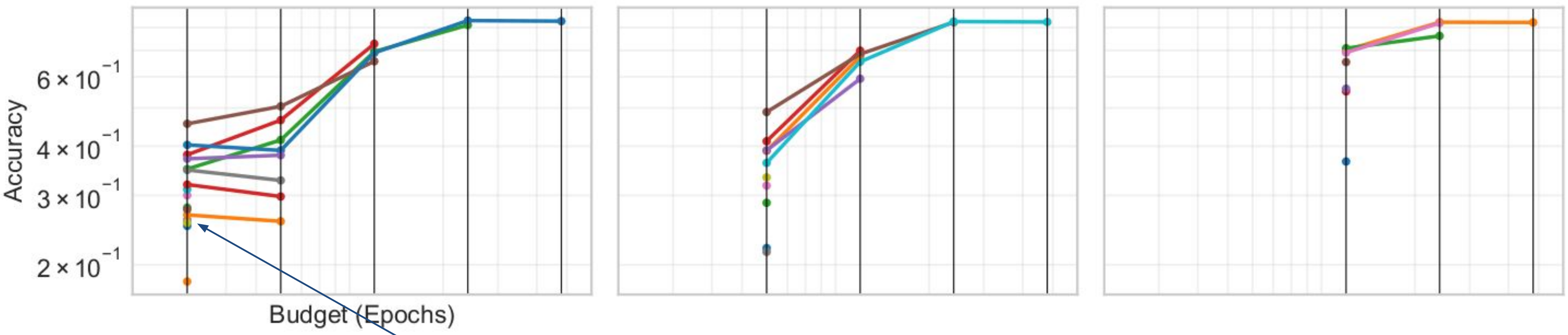
Idea: Do not fully train the lowest promising ones!

[AutoML Course]

Approach - Performance Estimation Strategy

Hyperband runs multiple copies of Successive Halving in parallel

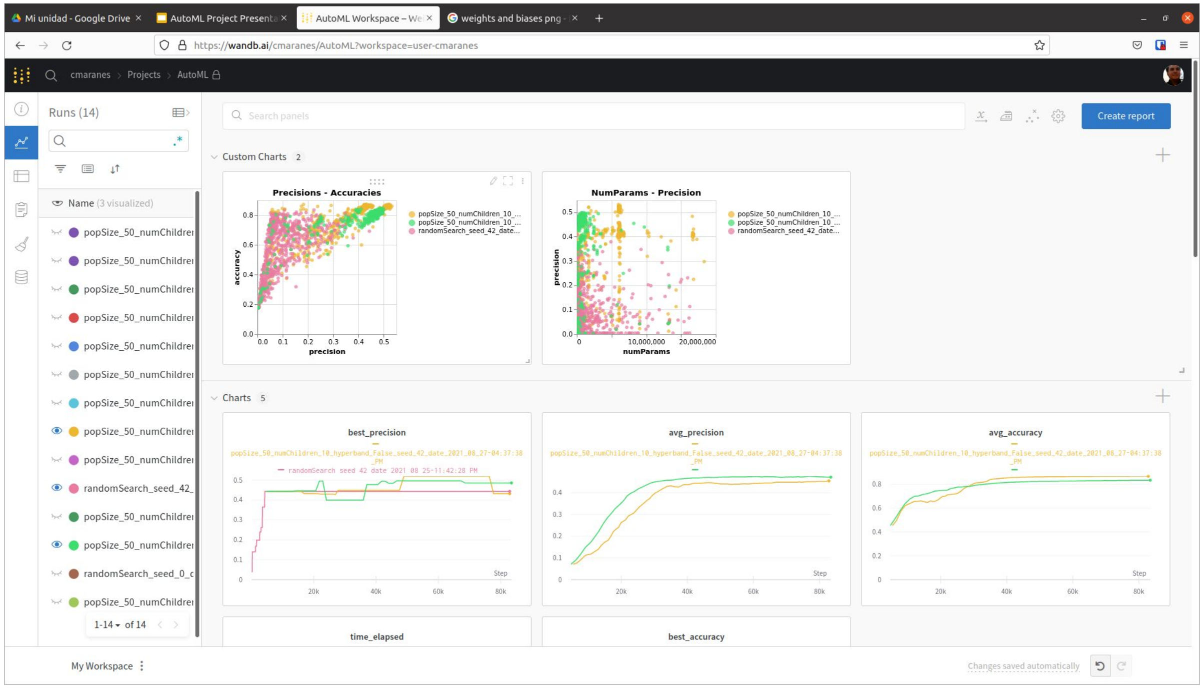
- Due to hardware limitations, it has been run in series



This is preferred since precision is better!

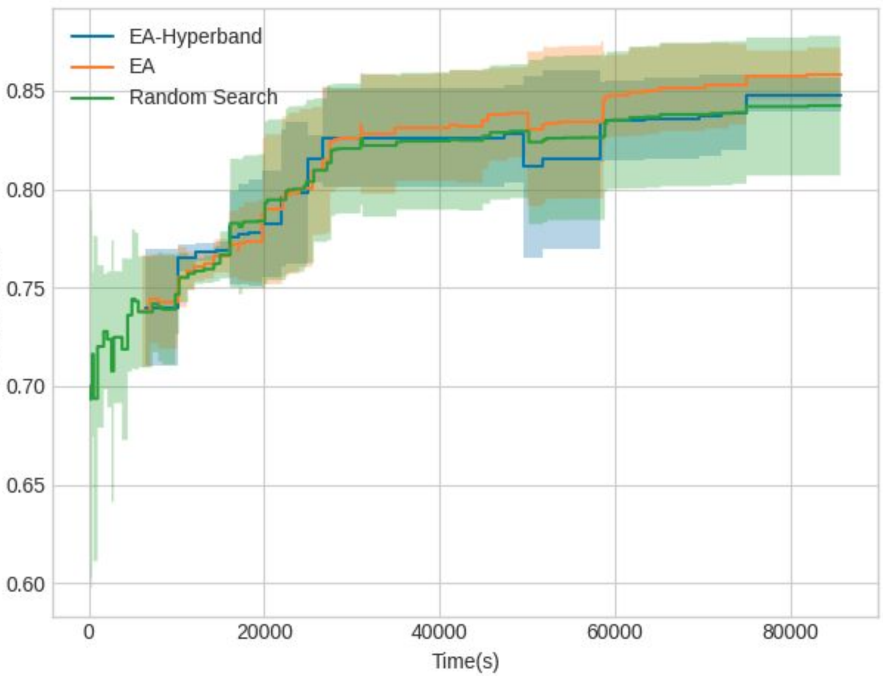
Training setup

- The algorithm has been run for 24 hours
- It has been run on the Pool Computers:
  - Intel(R) Core™ i7-3770 CPU @ 3.40GHz
  - Nvidia GeForce GTX 1060 3GB
- Weights & Biases integration
  - Real-time monitoring of the optimizer

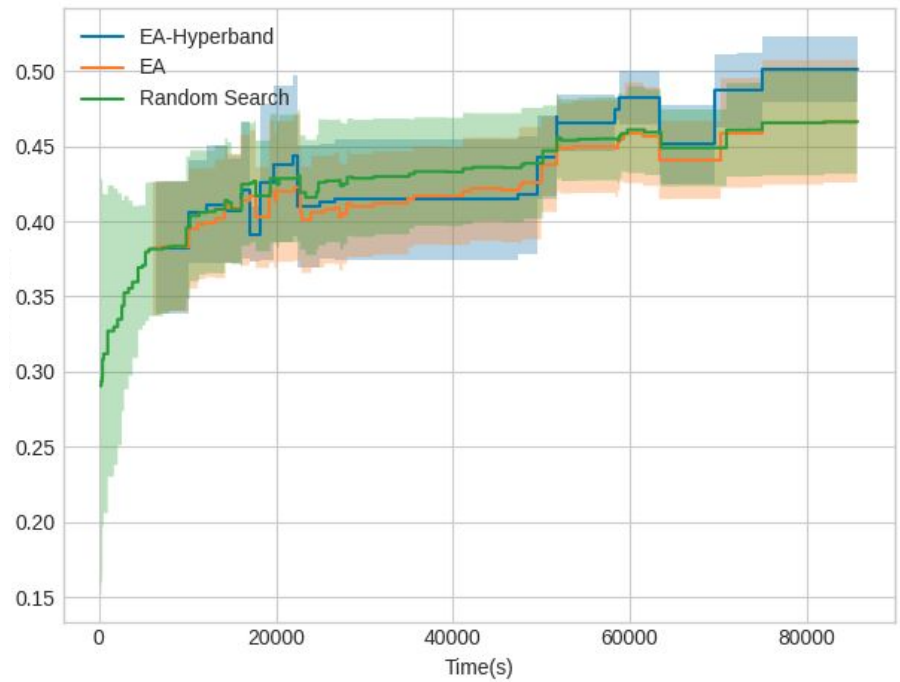


Results

Best Accuracy



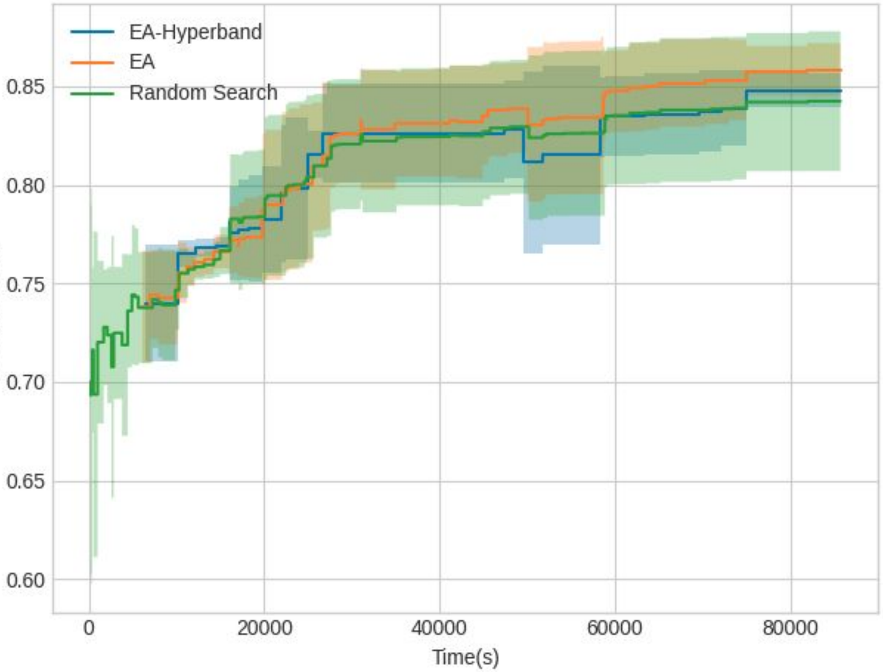
Best Precision



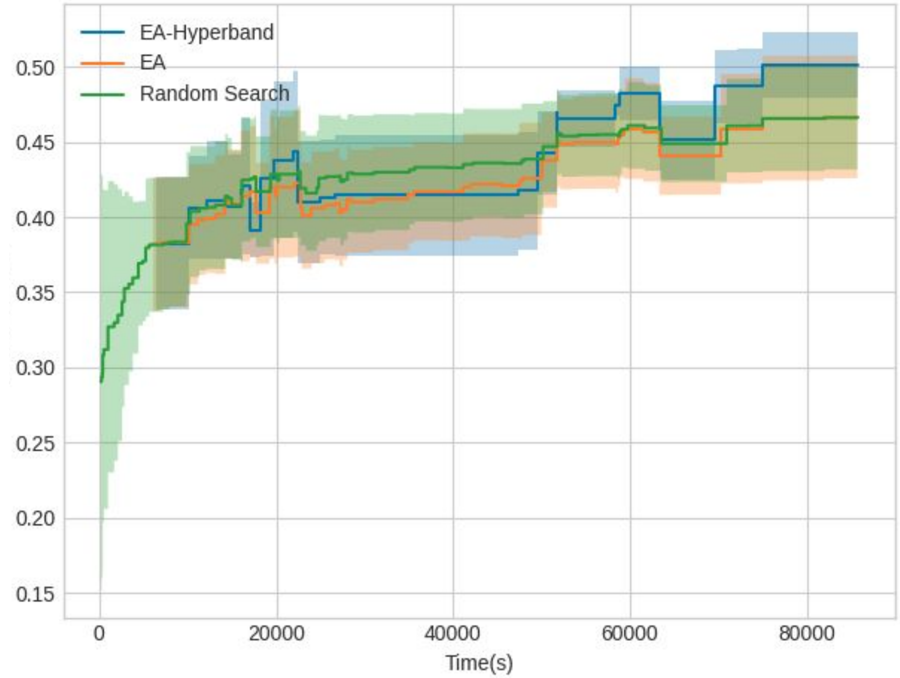
- Maximum number of parameters: 2e7
- Minimum precision: 0.39
- Tested random seeds: 0, 42, 123
- Random Search does not evaluate configurations with an invalid number of parameters

Results

Best Accuracy



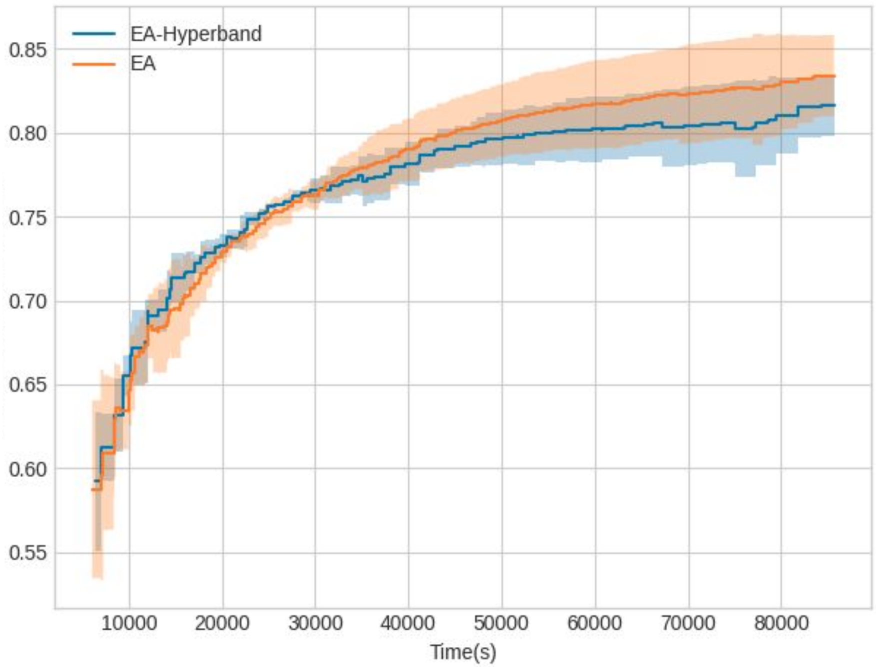
Best Precision



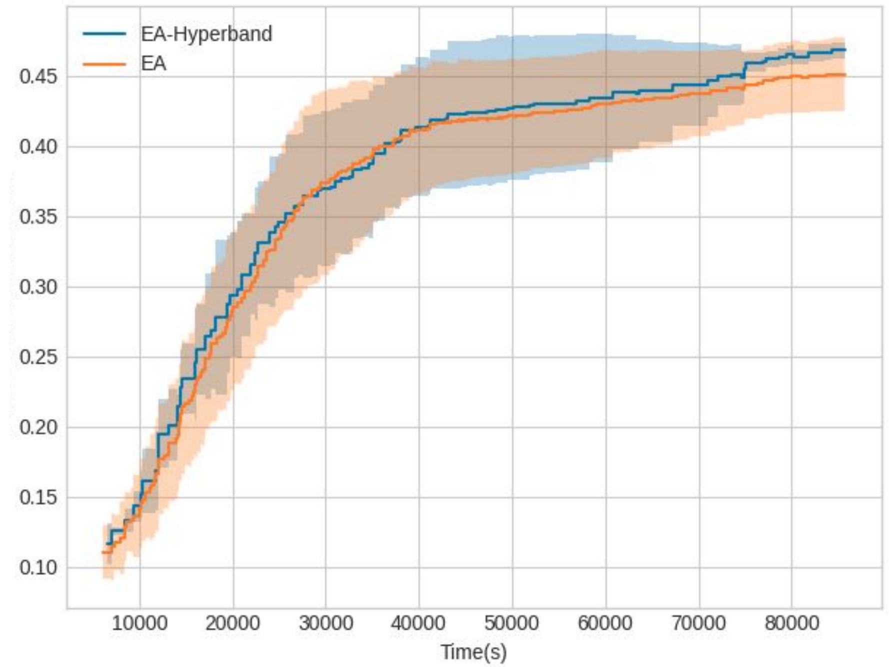
- All of them return a valid configuration
- Random Search returns the worst accuracy
- **EA-Hyperband** evaluates 44.46 (±17)% valid configurations
- **EA evaluates** 45.86 (±20)% valid configurations
- **Random Search** evaluates 0.7 (±0.4)% valid configurations

Results

Population average accuracy



Population average precision



- Fitness of the population improves over time
- **EA population** has a **better accuracy**
- **EA-Hyperband population** has a **better precision**

Results - Evaluation

Training and evaluation with test set:

	Precision	**Top3-Accuracy**
EA-Hyperband	0.552	0.891
EA	0.555	0.891
Random Search	0.569	0.876
Default	0.318	0.755

Results - Evaluation

Training and evaluation with test set:

	Precision	Top3-Accuracy
EA-Hyperband	0.552	0.891
EA	0.555	0.891
Random Search	0.569	0.876
Default	0.318	0.755

McNemar Test (α = 0.05):

- **EA-Hyperband vs. Default** (73.78>3.84): EA-HB has a better performance than default
- **EA-Hyperband vs. Random** (6.56>3.84): EA has a better performance than random
- **EA-Hyperband vs. EA** (0.1<3.84): Cannot say anything about the performance

Additional Results

The algorithm has been tested with different constraint values

Seed	Minimum Precision	Test Precision	Maximum Number of Parameters	Configuration Parameters	Test Accuracy
123	0.39	0.573	5e7	12322877	0.906
123	0.39	0.575	1e8	12322877	0.915
123	0.40	0.556	2e7	2251076	0.924
123	0.42	0.566	2e7	1897658	0.921
123	0.39	0.568	4e6	2021579	0.891
123	0.39	0.492	2e6	871034	0.900

- **All of them return a valid configuration**

Conclusion

- Evolutionary Algorithms are powerful enough to solve a NAS problem
- The idea is that the population satisfy the constraints to then optimize the accuracy
- Hyperband is applied to avoid wasting resources
- Experiments show that the proposed algorithm outperforms Random Search and the default network
- Pipeline allows reproducibility of the results

References

- [AutoML Course] - https://learn.ki-campus.org/courses/automl-luh2021/
- [Elsken et al. 2019] - Neural architecture search: A survey
- [Awad et al. 2021] - DEHB: Evolutionary Hyberband for Scalable, Robust and Efficient Hyperparameter Optimization
- GitHub Repository: https://github.com/automl-classroom/automl-ss21-final-project-cmaranes

Automated Machine Learning
Summer Semester 2021
Final Project

Carlos Marañes

Backup Slides

Results - Reproducibility

The project satisfy best practices for releasing code and when comparing NAS methods:

1. Reported hyperparameters for the final evaluation pipeline
2. Reported random seeds. Same input, same output (even with DropOut!)
3. Fair comparisons. Same seeds, same dataset, same hardware, same time, comparison with random search, etc.
4. GitHub Repository with detailed README

README.md

- target.py
  Target function. Is a handle of the CNN.
- utils.py
  Contains helpers for the different scripts.

How to evaluate the optimal found solution

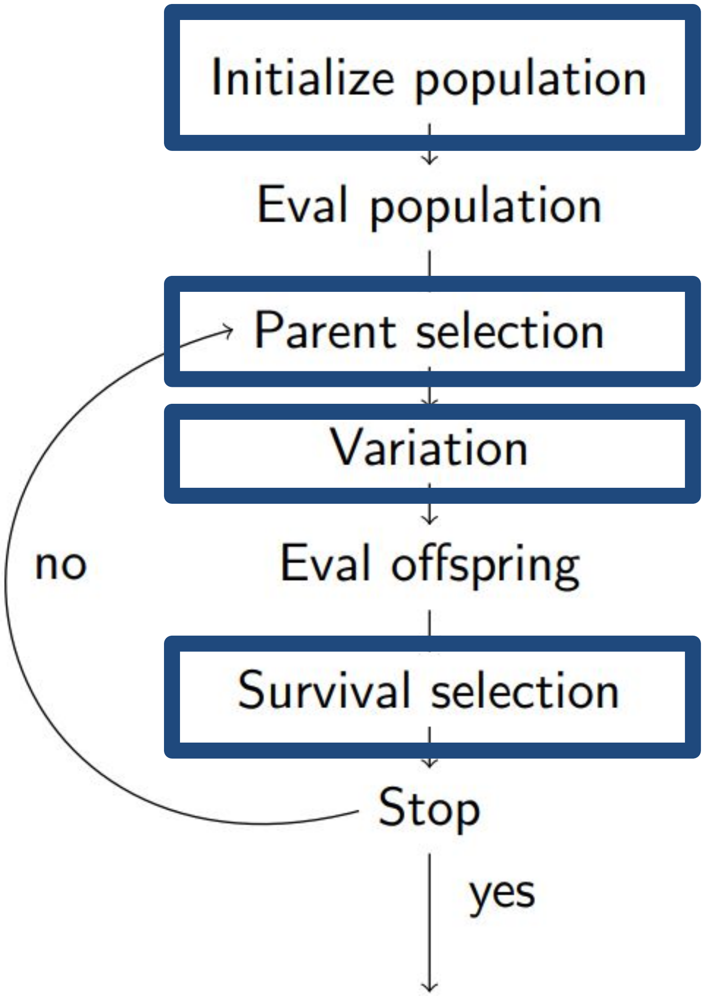
Once the optimizer has been run, it will output a file with the optimal configuration, which has the name of **opt_cfg_{moreInfo}**, where **moreInfo** indicates some information about the hyperparameters of how the algorithm has been run. This more info includes the random seed which can be passed as a parameter in --seed to obtain comparable results.

This file should be passed as the **--configuration** parameter to main.py to be evaluated

In this case, the best configuration obtained for the EA-Hyperband would be run with the following command:

```
$ python main.py -c
runs/opt_cfg_popSize_50_numChildren_10_hyperband_True_minPrecision_0.39_seed_123_date_2021_09_02-04:36
:25_PM.json -t True
```

Approach - Search Strategy



50 random selected individuals

Tournament parent selection

Mutation or recombination with a random probability of 0.5

(μ + λ)-selection

It selects from the 50+10 generated children the 50 best

[AutoML Course]

Approach - Hyperband Configuration

Hyperband Configuration:

- 5 Successive Halving runs
- The number of children is selected according to the budget of the corresponding Successive Halving
- Only fully trained offsprings are added to the population
- Hyperband warranties that at least is as bad as the Random Search
- A total of 10 new fully trained children are included into the population

Default Configuration

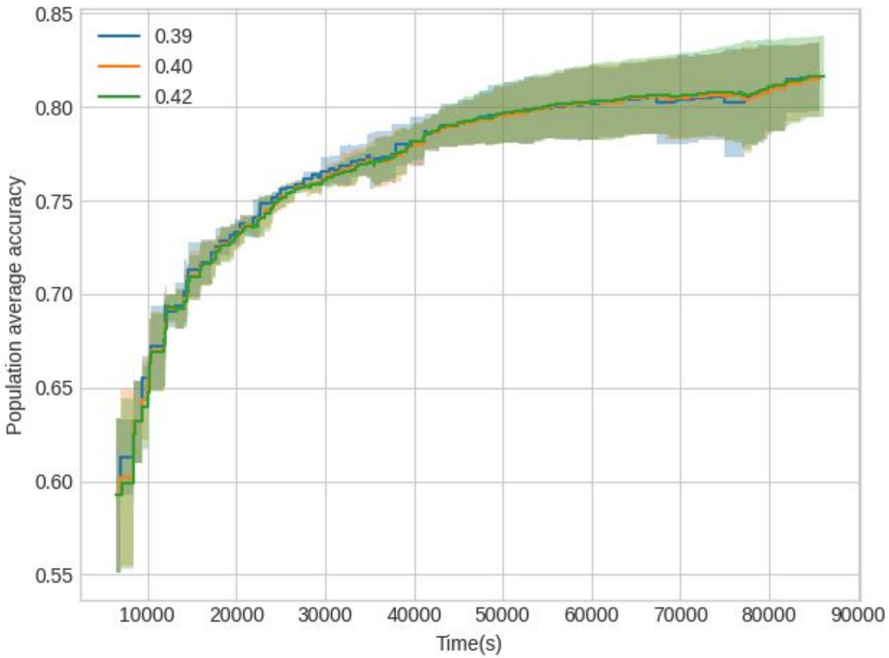
```
architecture_default = {
    'n_conv_layers': 3,
    'n_channels_conv_0': 457,
    'n_channels_conv_1': 511,
    'n_channels_conv_2': 38,
    'kernel_size': 5,
    'global_avg_pooling': True,
    'use_BN': False,
    'n_fc_layers': 2,
    'n_channels_fc_0': 27,
    'n_channels_fc_1': 17,
    'n_channels_fc_2': 273,
    'dropout_rate': 0.2}
```

Batch Size: 290

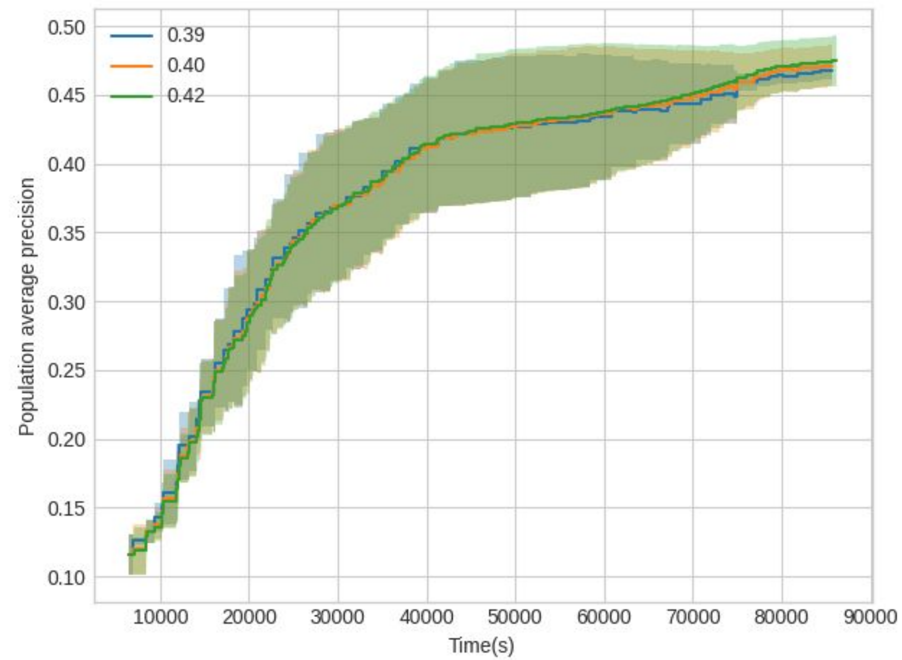
Leraning Rate: 0.0016

Results - Precision Constraint

Population average accuracy



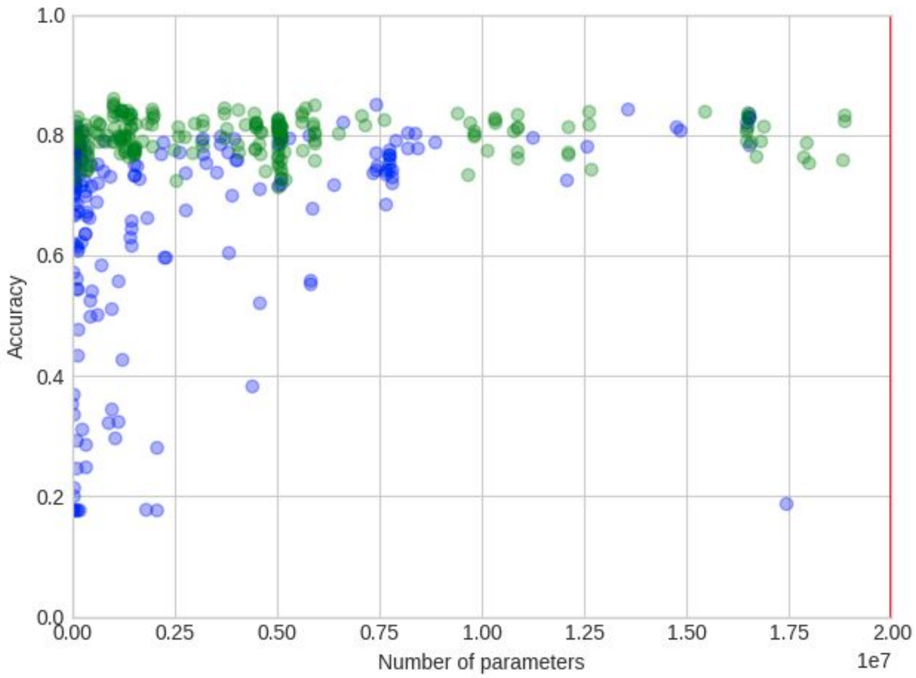
Population average precision

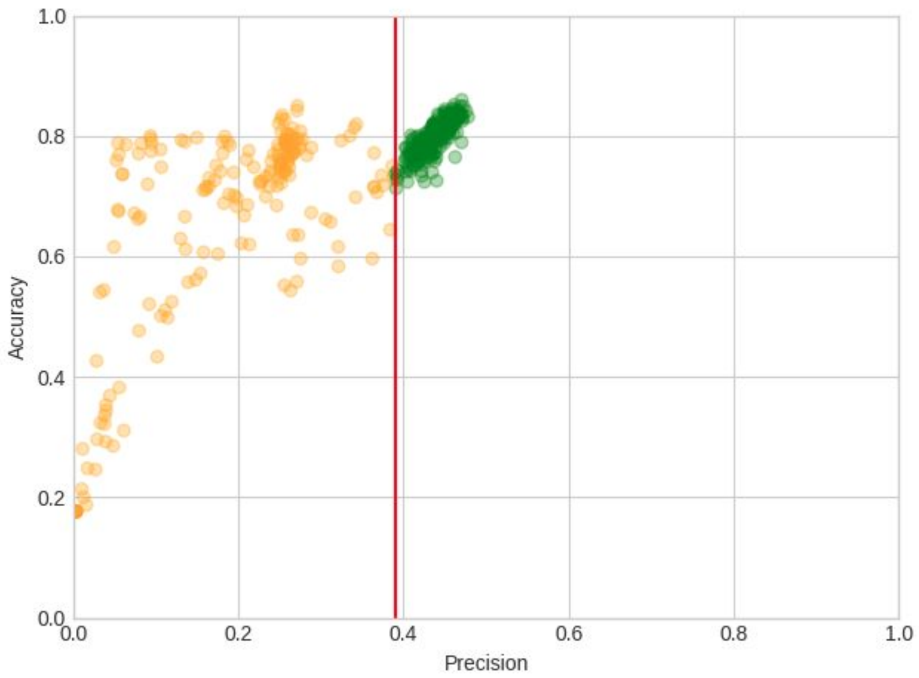


- Tested the algorithm with minimum precisions of 0.39, 0.40 and 0.42
- **All of them return a valid configuration**

Additional Results

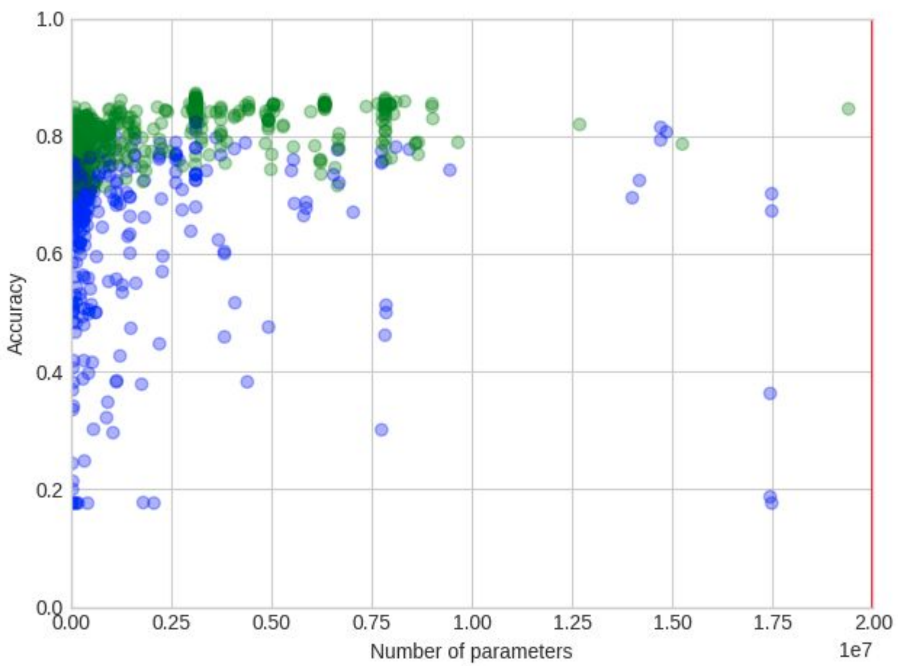
EA - Hyperband (Seed 123)

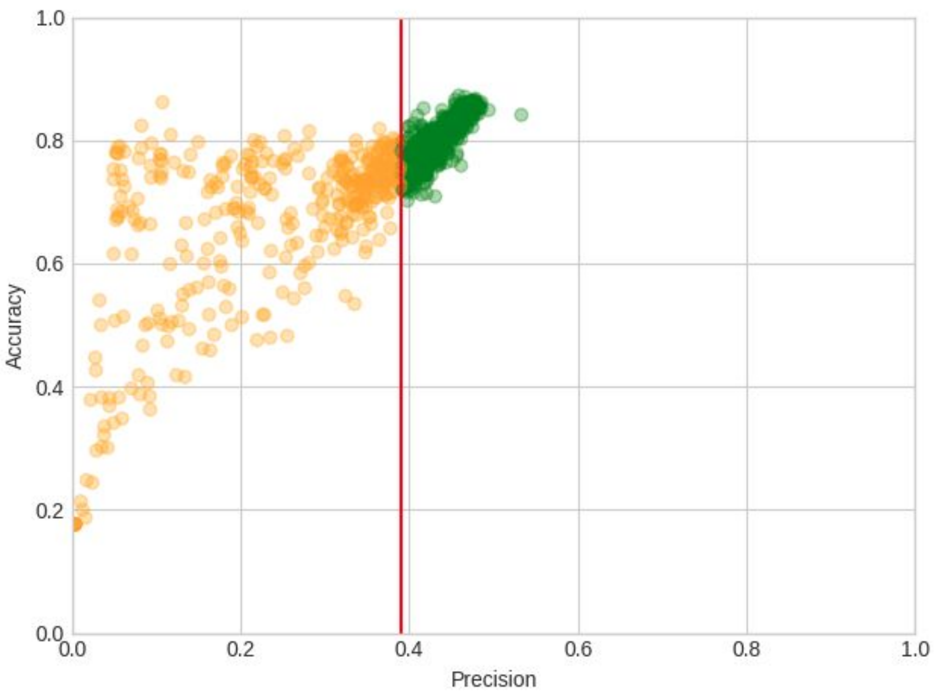




Additional Results

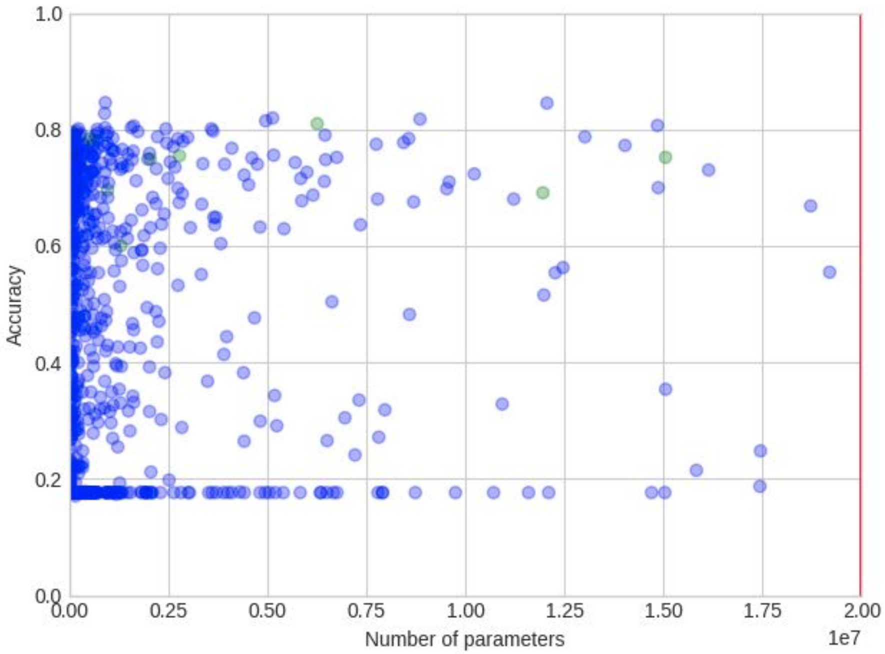
EA (Seed 123)

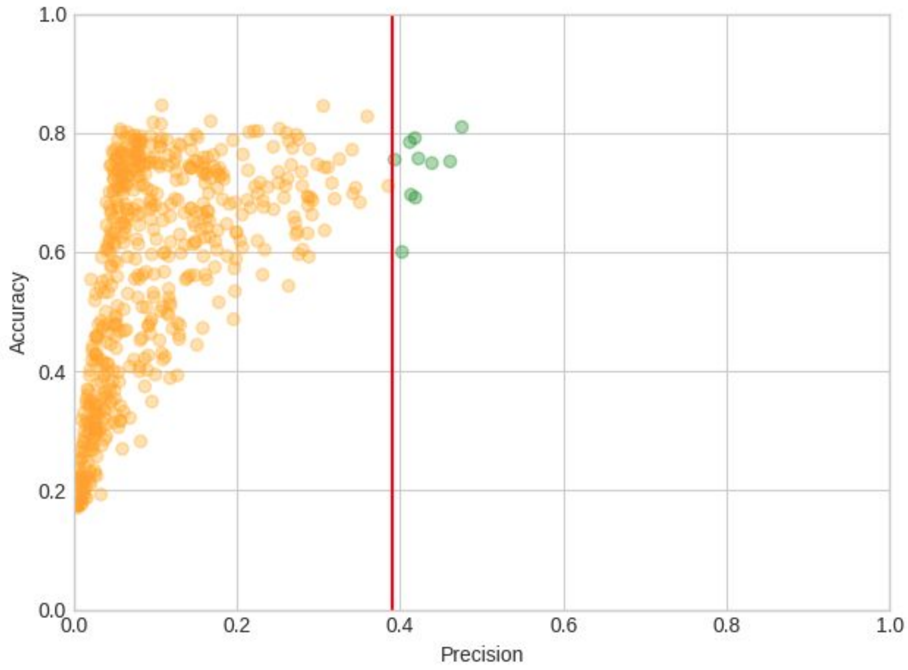




Additional Results

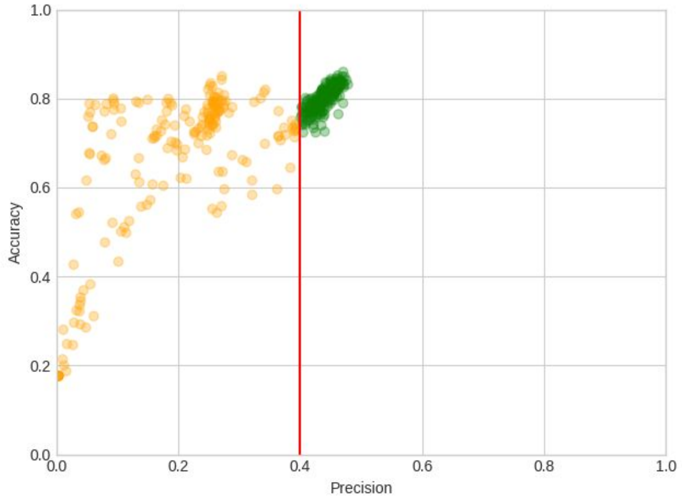
Random Search (Seed 123)



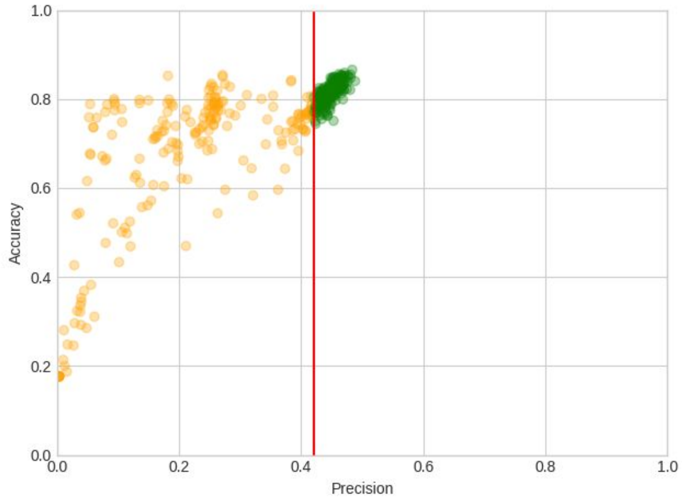


Additional Results - Seed 123

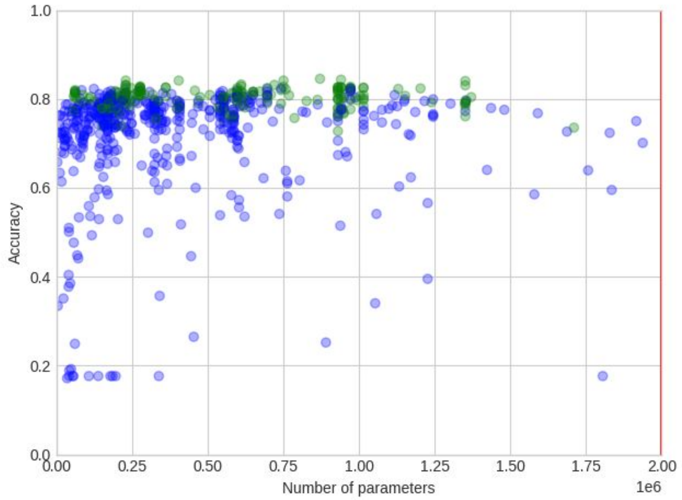
EA-HB - Prec: 0.40



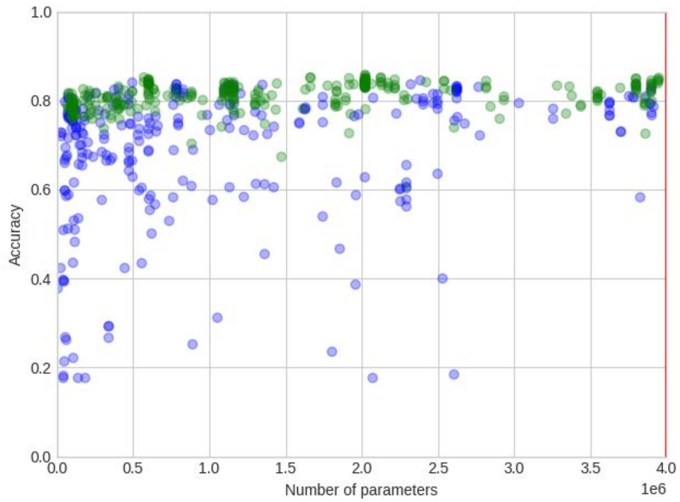
EA-HB - Prec: 0.42



EA-HB - Params: 2e6



EA-HB - Params: 4e6



Approach - Search Strategy

Lots of different ways of defining the search strategy stage arise thanks to the AutoML community. We can find:

1. Purely NAS Approaches
   a. Need of fully redefine the given code.
2. (HPO) Bayesian Optimization Strategies
   a. Surrogate Model: Random Forests allow the use of conditional and categorical search spaces.
      i. Poor uncertainty estimates.
      ii. Needs a significant amount of data to train the model.
3. (HPO) **Evolutionary Algorithms**
   a. Simple and Powerful [Awad et al. 2021]
   b. <u>Idea</u>: Achieve a population that satisfies the constraints and then improve accuracy

Approach - Search Strategy

Two constraints:

1. Maximum number of parameters
   a. Cheap to evaluate
   b. Discard all the individuals which do not meet this constraint
2. Minimum precision
   a. Expensive to evaluate
   b. Needed of evaluating to know if a certain individual meets this constraint

All members of the population satisfy the maximum number of parameters but not minimum precision

Some problems I have faced

- Some configurations of the search space could not be evaluated in the pool computers
- Disk Quota has been a problem when stopping and resuming training in Hyperband
- Some ConfigSpace functions did not return what expected