Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics

L. Tani†

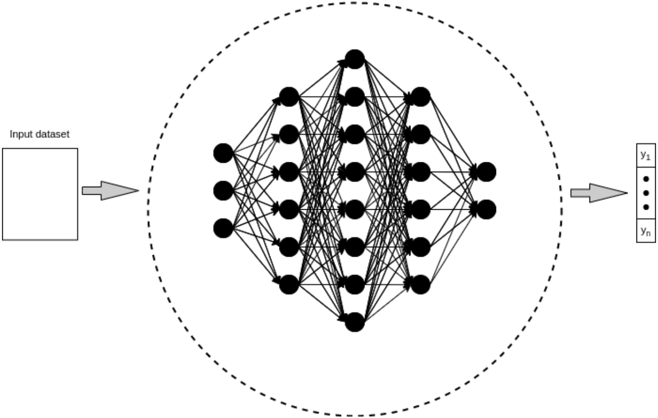
† *National Institute of Chemical Physics and Biophysics (NICPB)*

Oct. 10 - Oct. 12

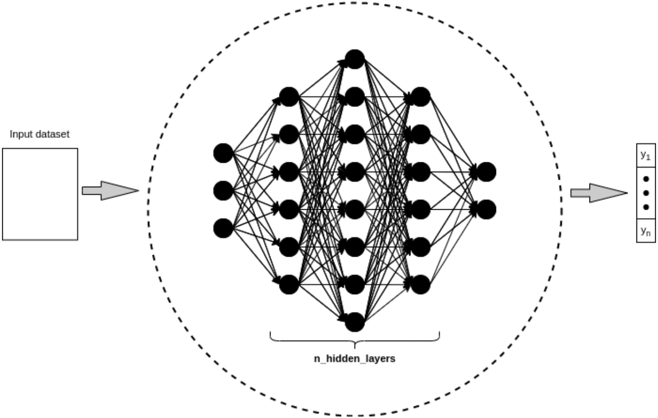
Vilnius, Lithuania

The 2nd CERN Baltic Conference

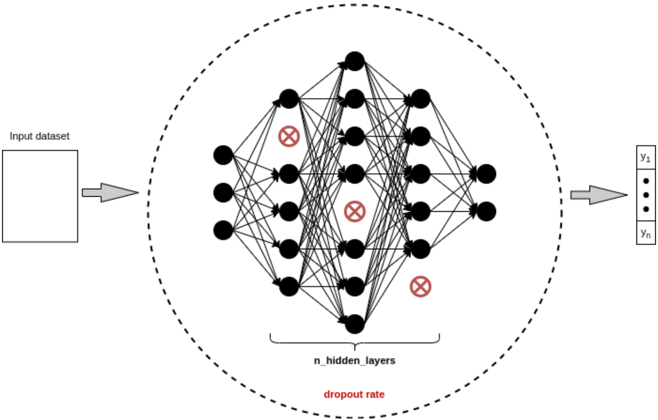
Motivation



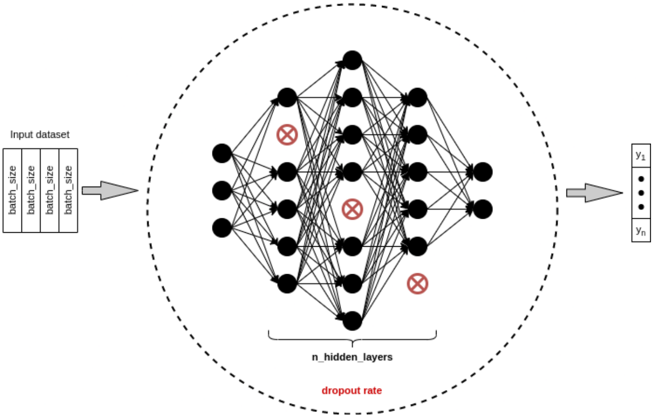
Motivation



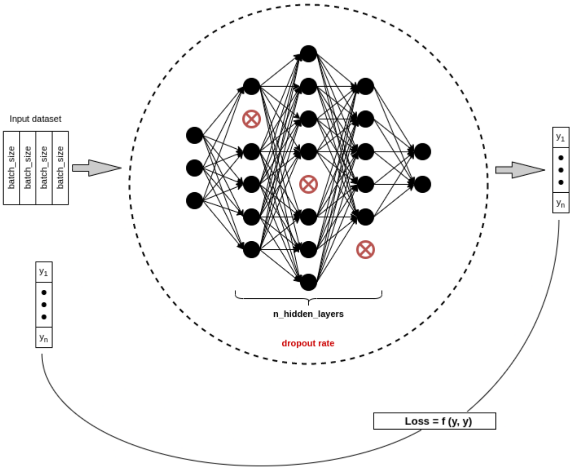
Motivation



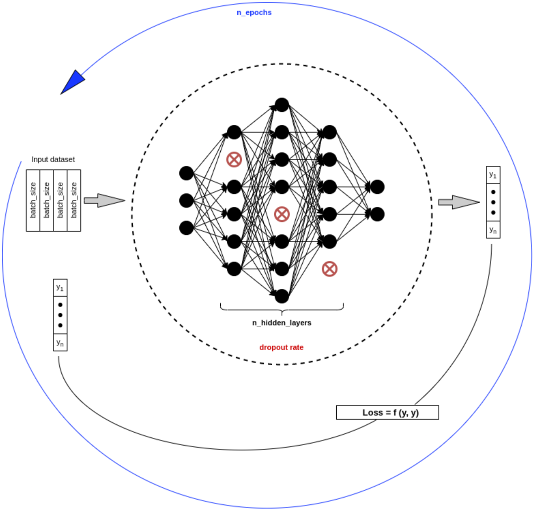
Motivation


Input dataset
batch_size
batch_size
batch_size
batch_size
n_hidden_layers
dropout rate
y_1
y_n

Motivation


Input dataset
batch_size
batch_size
batch_size
batch_size
n_hidden_layers
dropout rate
y1
yn
y1
yn
Loss = f (y, y)

Motivation

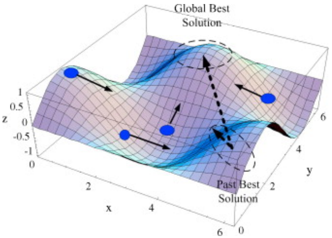

n_epochs
Input dataset
batch_size
batch_size
batch_size
batch_size
n_hidden_layers
dropout rate
y_1
y_n
y_1
y_n
Loss = f (y, y)

Motivation

- Significant impact on performance
- Manual hyperparameter optimization
- Automation
- HH $\rightarrow$ multilepton [Karl, Norman]
- Choice of best strategy unclear
- Parallelization @ HPC

Particle swarm optimization

- Swarm of particles
- Location = one solution
- Each value on an axis corresponds to one hyperparameter
- 3 steps of evolution:
  a **Espionage**

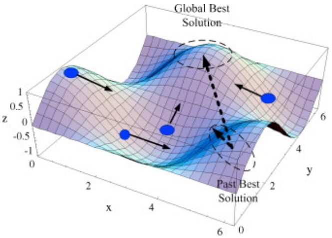


$$\mathbf{x}_{gb} = \operatorname{argmax}\{[\mathbf{x}^s_{pb} \in_R \mathcal{S}]^{(N_{info})}\}$$

Particle swarm optimization

- Swarm of particles
- Location = one solution
- Each value on an axis corresponds to one hyperparameter
- 3 steps of evolution:
  a Espionage
  b **Position update**

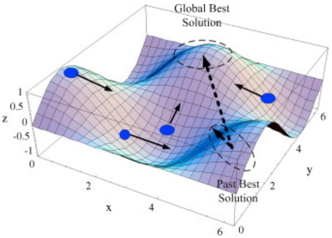
$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + w \cdot \mathbf{p}_i^k + \mathbf{F}_i^k$$



Particle swarm optimization

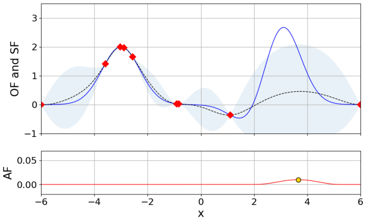
- Swarm of particles
- Location = one solution
- Each value on an axis corresponds to one hyperparameter
- 3 steps of evolution:
  - a Espionage
  - b Position update
  - c **Speed update**

$$\mathbf{p}_i^{k+1} = \mathbf{x}_i^{k+1} - \mathbf{x}_i^k$$



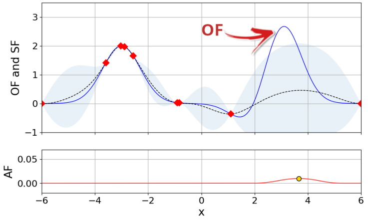
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



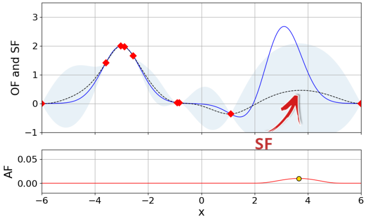
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



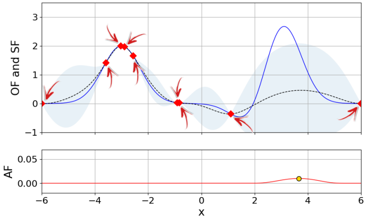
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



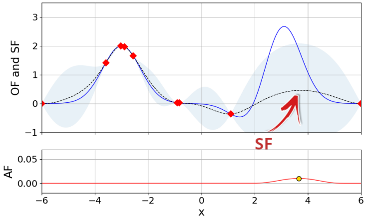
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



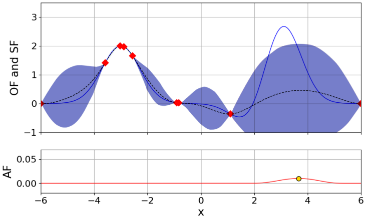
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



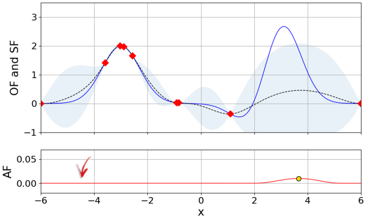
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



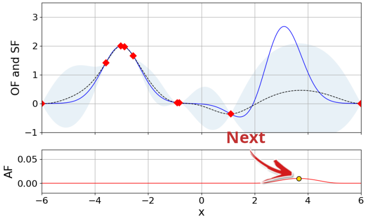
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



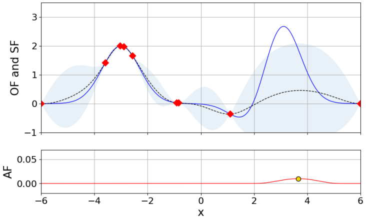
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



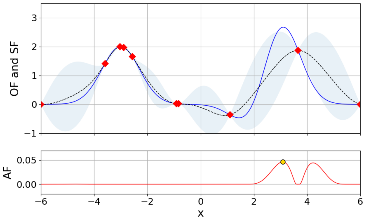
Bayesian optimization

- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



Bayesian optimization

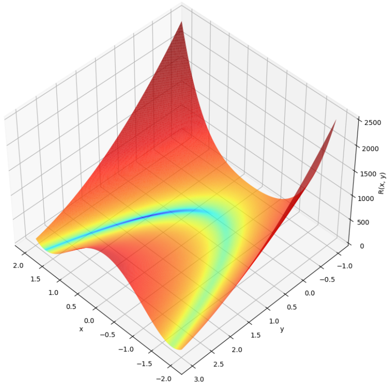
- Optimization done on surrogate function
  - Fast to evaluate
  - Derivatives and analytic form known
- Reported to work best with <1k evaluations
- 3 steps of evolution:
  - Find points to evaluate (q-EI)
  - Evaluate points
  - Update surrogate function



Rosenbrock function

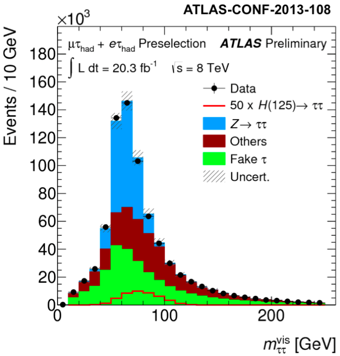
$$R(x, y) = (a - x)^2 + b(y - x^2)^2$$

- Well known trial function
- $(x, y)_{min} = (a, a^2)$
- Objective function
  Rosenbrock function itself.



ATLAS Higgs boson machine learning challenge (HBC) (i)

- Kaggle competition
- Run-1 simplified ATLAS $\mathrm{H} \rightarrow \tau\tau$ analysis
- Signal: $\mathrm{H} \rightarrow \tau\tau$
- Backgrounds:
  - $\mathrm{Z} \rightarrow \tau_h \tau_h$
  - $\mathrm{t\bar{t}} \rightarrow \tau_h + \mu/\mathrm{e}$
  - W-boson decay
- More representative task of ML in HEP analysis



ATLAS Higgs boson machine learning challenge (HBC) (iii)

Table: The seven chosen XGBoost hyperparameters to be optimized

	min	max
num-boost-round	1	500
learning-rate	10^{-5}	1.0
max-depth	1	6
gamma	0.0	5.0
min-child-weight	0.0	500.0
subsample	0.8	1.0
colsample-bytree	0.3	1.0

ATLAS Higgs boson machine learning challenge (HBC) (ii)

$$AMS = \sqrt{2 \cdot (s + b + b_r) \cdot ln[1 + \frac{s}{b + b_r}] - s}$$

ATLAS Higgs boson machine learning challenge (HBC) (ii)

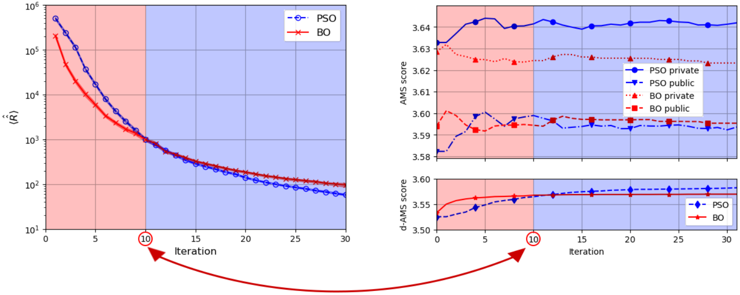
$$AMS = \sqrt{2 \cdot (s + b + b_r) \cdot ln[1 + \frac{s}{b + b_r}] - s}$$

$$\downarrow$$

$$dAMS = AMS_{test} - \kappa \cdot max(0, [AMS_{test} - AMS_{train}])$$

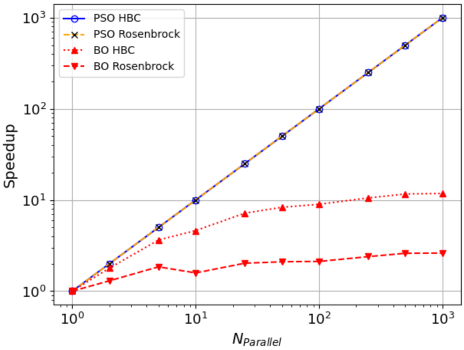
Performance

Rosenbrock function **HBC**

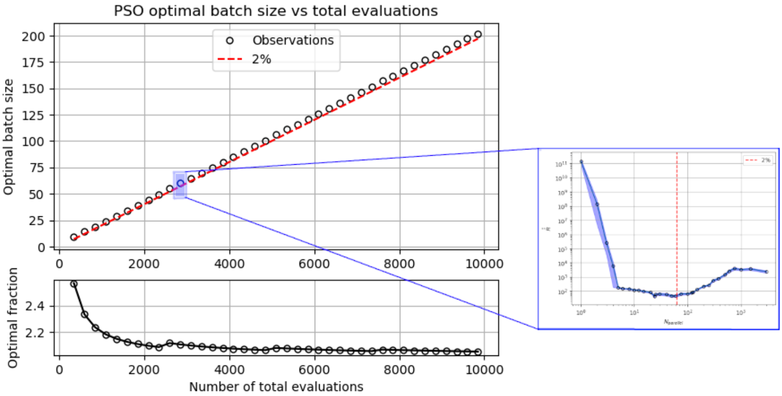


Parallelization (Amdahl's law)

$$S_{latency}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$



Parallelization (PSO)



Performance summary

	PSO	BO
Faster convergance	later	earlier
Parallelization capabilities	✓✓	✓
Suitable for low resources	✓	✓✓
Computational overhead	✓	?
Optimal $N_{parallel}^{relative}$	$\sim 2\%$	$<$

HH $\rightarrow$ multilepton: $\sim$10% improvement

Backup

References

L. Tani, D. Rand, C. Veelken, and M. Kadastik, "Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics," *The European Physical Journal C*, vol. 81, no. 2, pp. 1–9, 2021.

L. Tani and C. Veelken, "Comparison of bayesian and particle swarm algorithms for hyperparameter optimisation in machine learning applications in high energy physics," *arXiv preprint arXiv:2201.06809*, 2022.

title

Rosenbrock

	Time
Bayesian optimization + HBC ($N_{iter} = 30$ & $N_{parallel} = 100$)	3000 CPUh
Rosenbrock	0.06 CPUs
Particle swarm optimization	0.01 CPUs
HBC	$\mathcal{O}$(30 CPUmin)