

Weekly update

Christian Kirfel

19th of August 2020

Teaching

- Done! :)

QT updates

- Created several comparison presentations last week
- Improved plot quality, now using the up to date comparison script
- Updated tutorial and moved it to a repo (now with the rare feature of actually working)

Motivation

- Speed up the tedious process of hyperparameter optimisation
- Overcome the issue of optimal architecture being highly problem dependent
- Avoid biased decision of an expert user
- Utilize the baf job submission structure to efficiently run large samples of small optimisation jobs

Theory

- Code inspired by [1]
- Use a $(\lambda + \mu)$ scheme.
- λ individuals per generation
- μ individuals selected to create the next generation

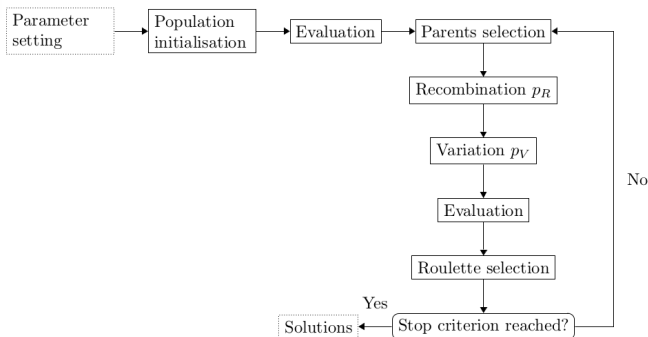


Figure 2: Breeding pipeline in EvoDeep algorithm.

Baf setup

- ① A bash script submits a badge of jobs from the local machine
- ② The script waits for all jobs to finish
- ③ The jobs are evaluated based on AUC and Accuracy
- ④ Based on the evaluation, a new set of jobs is created
- ⑤ Remnants of the old submission are removed to avoid exceeding quota

Early results

- Testing tZ versus $t\bar{t}b\bar{b}$
- Tested for Lorentz-Invariant and basic kinematic variables
- Testing on nodes, layers, learning rate, dropout
- The network ends around:
- Testing for similarity of final parameters. Even for small test samples the parameters are very similar
- Plots soon to come, unfortunately I had some plotting issues that are now fixed

Conclusion and future steps

- Running smoothly on baf for all test runs
- The network is approaching a reasonable order of magnitude and getting closer every single step
- Unreasonable genotypes are discarded
- Biggest impact made by epochs and metrics
- Weight ini has to be checked
- Starting parameters have to be set. Lately some runs resulted in bad parameters

Sources



Alejandro Martín García et al. “EvoDeep: a new Evolutionary approach for automatic Deep Neural Networks parametrisation”. In: *Journal of Parallel and Distributed Computing* 117 (Oct. 2017). DOI: 10.1016/j.jpdc.2017.09.006.