



kitchenSink.ai

The only way to claim SOTA is to benchmark against the rest

Introduction

Fractionated ML

The ML field is moving too rapidly. There is little confidence to be found in SOTA claims (e.g., in medicine).

Team

Samuel Marks

Academic

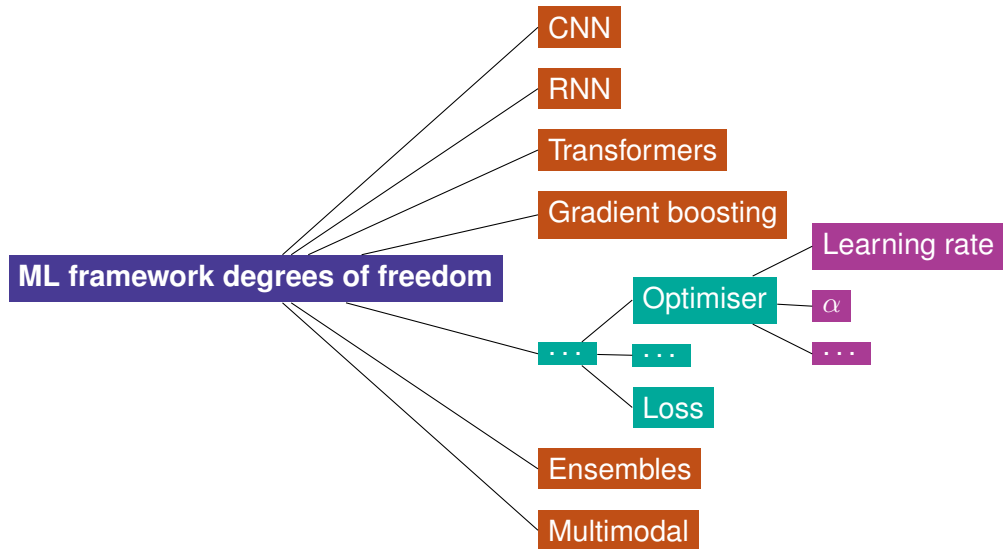
PhD from University of Sydney. On 2nd PhD at UNSW. Researcher at HMS/MEEI.

Commercial

Delivered many dozens of projects for dozens of companies over 10+ years.

Charitable

Working on facilitating mass screening programmes for blinding eye diseases.



Multi-ML

NOTE: The landscape is changing. I am a top 10 contributor to Google's [TensorFlow] Keras. A month ago they released a cross-compatible version that makes interchangeable: PyTorch, JAX, and TensorFlow.

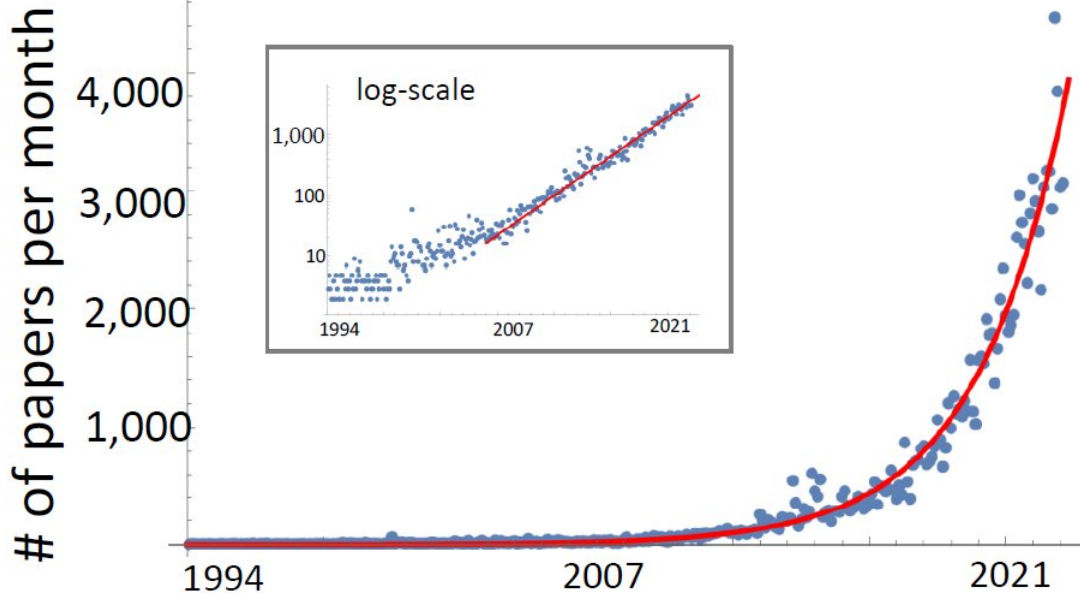
Multi-ML

There are 10 commonly used ML frameworks. Each have different ecosystems, and when a new research paper—or industry project—is released, they (usually) target just one framework.

My new multi-ML framework is created by applying my `cdd-python` compiler to 10 different popular ML frameworks (at the source-code level). This exposes CLIs, REST APIs, database tables and other useful layers.

Now, given a problem (e.g., determine best dataset for my new optimiser, or determine best [AUCROC] model for my new dataset), the framework will optimise across a *search space* traversing permutations of parameters (e.g., optimiser, loss function) and hyperparameters (e.g, α , β , learning rate). Where *search space* can include everything that the ML ecosystem has to offer.

ML+AI arXiv papers per month



Multi-ML - Future work $\frac{I}{II}$

Take arbitrary repos with Python packages or simple Notebooks & automatically:

0. Find and clone candidate repositories (e.g., from the arXiv);
1. Make OS independent;
2. Remove absolute paths (e.g., to weight files);
3. Format and autolint;
4. Add type hints;
5. Separate steps to be compatible with ensemble use-cases, e.g., move the model construction to its own function, and constants [like kernel sizes] to a consistent section;
6. Send pull-request / merge-request to repository;
7. If PR is accepted, add new model, optimizer, loss function, or other relevant 'thing' to this multi-ML meta-framework's search-space;
8. Publish (online) benchmarks of this new 'thing' against similar 'thing's on a variety of different datasets.

Multi-ML - Future work $\frac{//}{//}$

0. Automatically systematically review articles [with public datasets] coming through different research fields;
1. run their claims against the entire search-space of this multi-ML meta-framework;
2. (if improvement detected) write up a research paper with a new claim.
3. (if improvement detected) Open-source the (new) result with clear replication steps.

Add support for new ML frameworks.

Analytics and dashboarding.