Summary of Machine Learning + Differentiable Programming @ Paris

Nathan Simpson

HSF Analysis Ecosystems, 25/05/22

LUND UNIVERSITY

ATLAS EXPERIMENT

Differentiable Programming

People are interested in diffprog!

- Many people in the room (at least for the second session) expressed a curiosity in this new paradigm
- Use cases are slowly arising:
    - **Speeding up fitting with automatic gradients**
    - **End-to-end summary statistic optimisation**
    - **Calibration of data by tuning a differentiable simulator**
- Still remains largely unused, picked up mostly by the ML-savvy (but this can change)

large

Big question mark: <u>is it worth it?</u>

How does it compare to existing methods?

- Summary statistic-based methods, e.g. INFERNO, neos, have shown some improvement on toy problems and open data
- Also come with an additional factor of compute, and batching concerns (whole analysis in one batch?)
- Not many other uses with comparisons already done! [that came up today]

Can it scale? *We don't know!*

So one thing that is clear: <u>we need toy problems that have a degree of realism for comparison</u>, along with <u>expert-tuned benchmarks from standard methods</u>

The ecosystem idea for relaxation

Since we have a modular set of operations to make differentiable [histograms, cuts, fits], we could greatly benefit from a module which

- Provides drop-in replacements for non-differentiable operations
- Has a similar API to existing tools for ease of use
- Makes use of continuous integration to test
    - Validity of approximations
    - Ability to take gradients across operations
- Still needs people to have use cases (but seems like there may be some)

`relaxed` 😴

Machine Learning

Common, realistic benchmarks!

- Standard ML community has very established problems that are used as metrics for all new methods
  - Handwritten digits (MNIST), CIFAR-10, etc.
- We're taking steps towards making our workflows available to people for this
  - TrackML competition in 2018
  - Calorimeter challenge
  - Your idea here!
- Also could help consolidate training data for large models

Experiment tracking

Clear that we don't do enough of this (and it's not centralised)
- tools exist, like tensorboard, weights and biases, comet.ml

We could also use this in general for analysis optimisation!
- e.g. seeing how your cutflow opt is doing over time, tracking a bunch of principled metrics along the way

Platforms

Again, this is not a centralised thing (some people have their own GPU clusters, some don’t)

Some effort at CERN, e.g. ml.cern.ch, which provides a jupyter entrypoint to using GPUs + Kubeflow for sophisticated tracking of training workflows

- Also work on a VSCode frontend

Nice if more effort put into this kind of thing + visibility, especially for people still training on lxplus CPU [they exist, and we should end their suffering]

Large models (that are also big)

This is the **number one advancement of ML in industry, and we're not using it**

- Models trained at scale with many, many parameters are showing a degree of generalisation and performance that wasn't thought possible (particularly for language)

We didn't discuss it much, mostly because we have no experience with this type of thing. -> possible industry collaboration point?

Would require very carefully selected task definition, lots of training data, and a metric buttload of compute

Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butter-fly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

Thanks :)