Research Proposal

Thomas Foster

January 2023

1 Overview

Deep learning is helping mankind solve some of its greatest problems, from defeating long- standing diseases [8] to reducing the effects of climate change [4]. Of course, large neural networks are not perfect. A famous example comes from vision, where classifiers can be adversarially tricked into classifying Pandas as Gibbons[7]. Language models can be similarly tricked into producing inconsistent answers[2]. Further examples of this fragility can be seen in getting GPT3 to understand inferences[18], or watching stable diffusion fall apart with compositional prompts.

Discovering these mistakes is hard, and finding creative ways to trick models will get harder as AI progresses. As we apply AI to more nuanced tasks, judging whether a mistake has been made will be prohibitively time-consuming, let alone providing corrections at a scale useful for training.

However, just because we cannot find mistakes does not mean they won't exist. As AI improves, we will give it more responsibility, multiplying the impact of any mistakes. There will be much incentive to deceive models. The distribution of inputs seen in training is unlikely to match the adversarial distribution seen in production, especially if there are multiple competing AI agents.

My primary research goal is therefore to automate the evaluation of deep neural nets, developing techniques to evaluate models on complex criteria and systematically generate challenging examples. Ultimately I wish to turn these techniques into automated and open-ended training paradigms.

2 Technical directions

Currently, we train on simple tasks that we can perform at scale (e.g next token prediction in language modeling[1]) and it is only afterwards that we asses whether the higher-level concepts we care about (reasoning, reliability, consistency etc) have emerged. Learned reward functions offer an interesting alternative. I experienced this at at Genei[20], a startup I founded during my final year, where we trained a model to score summaries according to user preferences. By optimising against this reward model with reinforcement learning, we produced a summariser that users far preferred to the original.

Learned reward functions do present new challenges, notably that optimising against a poorly generalised reward model can yield spurious solutions that are assigned high rewards when they shouldn't be.

We can also use pre-trained models to help us test concepts like reliability and consistency. In [2] the authors show how, given a question about an image, models can be forced to generate inconsistent (answer, explanation) pairs. Their process is intricate and uses human labellers in several places, such as editing consistent explanations to be inconsistent, or deciding whether generated explanations are consistent. LLMs could perform both of these tasks, turning the author's process into a text-based RL environment. Using the number of consistent explanations as a reward, we could train a model to be more robust to this attack.

Another example of RL environments testing higher order capabilities is the TextWorld [5] environment, a parser-based game in which the player responds to text descriptions of a game state and completes missions. This tests language grounding and common sense reasoning - knowing not to tell the game to "climb an apple" or “eat a tree”. However, TextWorld is powered by context-free grammars, which limits the complexity of both game state descriptions and accepted agent responses. Using language models to do this would produce more realistic and challenging environments. I am excited about the wider use of language models in RL environments to specify goals [21] or provide qualitative feedback to an agent at the end of an episode.

Another major challenge is systematically finding examples that models get wrong. There is much important work into curriculum generation [16] and uncertainty quantification [15], but what I find most exciting is the prospect of learning optimisers [14] or RL algorithms [13] that do this without human design. For example, distilling active-learning algorithms [9] into transformer models could yield better in-context active learning. This seems within reach, with language models already being shown to be able to express their uncertainty in words [12].

With techniques to scalably evaluate models on what we care about, and systematically find challenging situations, we can turn them into RL training procedures. Further research is needed into the algorithms to solve rich, complex environments with large models. For example, to stabilise training, PPO [19] requires 3 copies of the policy to compute KLD estimates, in addition to a critic network and potentially a learned reward function. This is very expensive when networks have billions of parameters. Further, RL on language uses a large action space (the size of the language vocabulary), which can make learning the value of states and actions challenging and sample intensive. It also exacerbates errors in the KLD estimates, resulting in more unstable training.

Some argue that gradient-free or evolutionary approaches (such as ELM [10]) are more promising than optimising for desirable properties directly. Regardless of how intelligent systems are produced, we will need ways to challenge their performance on concepts we care about.

3 My research background

In 2020, I graduated with 1st class honours and 2nd in my cohort from the University of Oxford’s 4 year Computer Science course. Whilst there, I developed my theoretical knowledge through taught courses, and developed strong practical skills, winning the GResearch prize for best 2nd-year group practical.

I undertook 2 extended research projects whilst at Oxford. During my 3rd year, I used approximate Bayesian computation to find probabilistic context-free grammar rules that best approximated 3D models. I advanced previous methods, where such rules had to be designed by hand, and earned a distinction.

For my master’s dissertation, I developed a novel algorithm for high dimensional integration. My method [6] was more accurate and used fewer samples than existing methods [17]. We generated some exciting discussions upon release of the work, such as a response paper [11] from Peter Lepage, who won the 2016 Sakurai prize in physics. I was awarded the Microsoft Prize for best undergraduate dissertation for this work.

These achievements required me to develop a lot of resilience. I was hospitalised for several months during exam term in my first year, after rupturing my intestine in a rugby match. On my return I sustained a major concussion that banished to a dark room for most of the winter term. It took until near the end of my 2nd year before I felt I had fully caught up.

Since graduating I have strived to do innovative work wherever I can, and I’ve also given up playing rugby. At Genei [20], I extended reinforcement learning for language models, by removing the dependence on explicitly given ratings, and learnt the reward model by in-app user micro-interactions.

Building my startup over the last 3 years has been immensely rewarding, but has lacked the research freedom for me to fully pursue the academic interests most important to me. In late 2022 I left to pursue research full-time, and I’m currently working on numerous research projects, such as with Carper.ai [3] to investigate RL algorithm distillation.

References

[1] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *CoRR* abs/2005.14165 (2020). arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[2] Oana-Maria Camburu et al. “Make Up Your Mind! Adversarial Generation of Inconsistent Natural Language Explanations”. In: *CoRR* abs/1910.03065 (2019). arXiv: 1910.03065. URL: http://arxiv.org/abs/1910.03065.

[3] Carper.ai. 2022. URL: https://carper.ai/ (visited on 12/01/2022).

[4] Climatetrace.org. *Independent greenhouse gas emissions tracking*. 2022. URL: https://climatetrace.org/ (visited on 12/01/2022).

[5] Marc-Alexandre Côté et al. “TextWorld: A Learning Environment for Text-based Games”. In: *CoRR* abs/1806.11532 (2018). arXiv: 1806.11532. URL: http://arxiv.org/abs/1806.11532.

[6] Thomas Foster et al. *Model Evidence with Fast Tree Based Quadrature*. 2020. DOI: 10.48550/ARXIV.2005.11300. URL: https://arxiv.org/abs/2005.11300.

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2014. DOI: 10.48550/ARXIV.1412.6572. URL: https://arxiv.org/abs/1412.6572.

[8] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://doi.org/10.1038/s41586-021-03819-2.

[9] Michael Laskin et al. *In-context Reinforcement Learning with Algorithm Distillation*. 2022. DOI: 10.48550/ARXIV.2210.14215. URL: https://arxiv.org/abs/2210.14215.

[10] Joel Lehman et al. *Evolution through Large Models*. 2022. DOI: 10.48550/ARXIV.2206.08896. URL: https://arxiv.org/abs/2206.08896.

[11] G. Peter Lepage. “Adaptive multidimensional integration: vegas enhanced”. In: *Journal of Computational Physics* 439 (Aug. 2021), p. 110386. DOI: 10.1016/j.jcp.2021.110386. URL: https://doi.org/10.1016%2Fj.jcp.2021.110386.

[12] Stephanie Lin, Jacob Hilton, and Owain Evans. *Teaching Models to Express Their Uncertainty in Words*. 2022. DOI: 10.48550/ARXIV.2205.14334. URL: https://arxiv.org/abs/2205.14334.

[13] Chris Lu et al. *Discovered Policy Optimisation*. 2022. DOI: 10.48550/ARXIV.2210.05639. URL: https://arxiv.org/abs/2210.05639.

[14] Luke Metz et al. *VeLO: Training Versatile Learned Optimizers by Scaling Up*. 2022. DOI: 10.48550/ARXIV.2211.09760. URL: https://arxiv.org/abs/2211.09760.

[15] Sören Mindermann et al. *Prioritized Training on Points that are Learnable, Worth Learning, and Not Yet Learnt*. 2022. DOI: 10.48550/ARXIV.2206.07137. URL: https://arxiv.org/abs/2206.07137.

[16] Jack Parker-Holder et al. *Evolving Curricula with Regret-Based Environment Design*. 2022. DOI: 10.48550/ARXIV.2203.01302. URL: https://arxiv.org/abs/2203.01302.

[17] Peter Lepage. *The Vegas algorithm*. 2022. URL: https://github.com/gplepage/vegas (visited on 12/01/2022).

[18] Laura Ruis et al. *Large language models are not zero-shot communicators*. 2022. DOI: 10.48550/ARXIV.2210.14986. URL: https://arxiv.org/abs/2210.14986.

[19] John Schulman et al. “Proximal Policy Optimization Algorithms”. In: *CoRR* abs/1707.06347 (2017). arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.

[20] Jack Bowen Thomas Foster and Adrien Wald. *Genei.io*. 2022. URL: https://genei.io (visited on 12/01/2022).

[21] Ted Xiao et al. *Robotic Skill Acquisition via Instruction Augmentation with Vision-Language Models*. 2022. DOI: 10.48550/ARXIV.2211.11736. URL: https://arxiv.org/abs/2211.11736.