DLAI projects

A.Y. 2023/24

Prof. Emanuele Rodolà - rodola@di.uniroma1.it
Dr. Donato Crisostomi - crisostomi@di.uniroma1.it
Dr. Robert Adrian Minut - robertadrian.minut@uniroma1.it
Dr. Daniele Solombrino - solombrino.1743111@studenti.uniroma1.it

Introduction

In this document, we present the projects for the Deep Learning exam in A.Y. 2023/24 🎆

Every project covers different aspects of the course, so make sure to select the one that catches the most of your interests.

Originality and creativity is appreciated and rewarded, so, if you want to go further than the minimum requirements listed for each project, you are more than welcome to do it!

Resources

Each project has a set of resources which may come in handy during the solution development. However, the list is not exhaustive and may not cover all your needs, so you are encouraged to look for additional resources online or in other project tracks proposed here.

Us

In case of need, please reach us at the four addresses listed above, using the subject **[DLAI 2023/24] project assistance**.

For bureaucratic or sensitive reasons, you can message just Prof. Rodolà.

Submission

To submit your project, send an email to the four addresses at the beginning of the document with the following subject: **[DLAI 2023/24] project submission**.

Attach the report and a git repository hosting the code you produced to the body of email.

Report

The report **must** follow the template available at this link:
https://github.com/erodola/DLAI-s2-2024/raw/main/template.zip

The goal of the report is to show us that you understand the concepts related to the project and to motivate all aspects of the project, starting from your line of attack, moving into implementation details and concluding with a comment on the obtained results.

Do not limit yourself to simply show facts, rather show us your reasoning about them and interpret them according to what we have seen in class or other topics you studied in other courses or on your own.

A list of projects follows below.

1. Machine unlearning

Can you ***unlearn*** something?

Your task here is the following: given a network pre-trained on some data, you want to finetune it to selectively forget a class, and learn a new class.

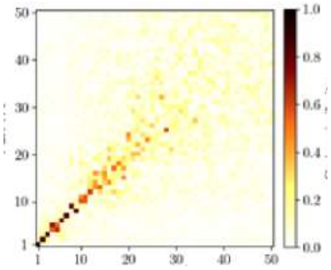
As an initial approach, you may do the following. Start with a MNIST classifier pre-trained on a subset of the digits. Now replace one of the learned digits, say the class “6”, with a new digit, say “3”. A possible way to proceed is to identify which weights are more involved in the prediction of class “6”, freeze all the rest, and train with a loss that favors the “3” while penalizing the “6”. Test this baseline and see whether it brings you anywhere. Are there any pitfalls in this idea? Does it work? Use it as a first line of attack to understand the problem.

Starting from these baseline tests, devise a new unlearning procedure. You can improve upon this baseline, make up your own idea from scratch, or check the literature to get ideas. If you use an existing approach, ***you must add something new***, for example by testing it on some new data modality (e.g., audio), by studying more extreme cases, failures, weaknesses, or by making it more efficient, and so on.

2. Spectral maps rediscovered

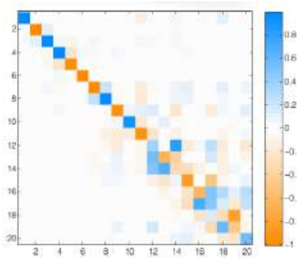
Have a look at this paper: https://openreview.net/forum?id=ofEBFOrITI

The paper introduces a technique to compare the trained weights of neural networks, giving rise to a new representation that is easy to interpret, and that is suggested to be universal across datasets. The representation is **spectral**, **easy to compute**, **matrix-based**, and looks like this:



Now look at this other paper from the graphics / geometry processing community: https://people.csail.mit.edu/jsolomon/assets/fmaps.pdf

The paper introduces a new representation for finding correspondences between different 3D shapes. Just like in the previous case, the representation is **spectral**, **easy to compute**, **matrix-based**, and looks like this:

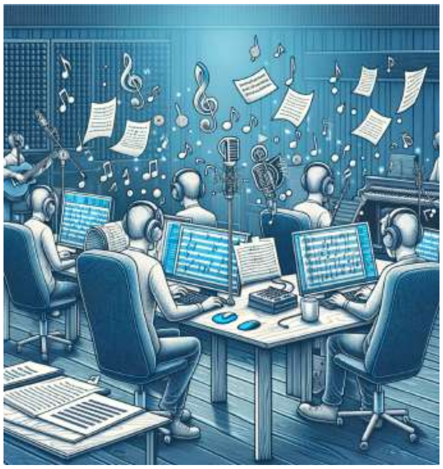


Both representations are based on computing inner products between eigenvectors. This looks like a re-discovery of the same phenomenon by different communities and on the different data -- or is it?

For this project, you will investigate the similarities and differences between the two representations and explain how one relates to the other. If the two representations are actually equivalent, you must provide a mathematical proof. If they are different, you must demonstrate this through examples, showing the benefits and drawbacks of one over the other.

Bonus points if you can find new applications of either representation! Examples include but are not limited to: a new measure to quantify network similarity, a new way to merge the weights of different networks, a new way to perform network stitching (e.g. use an encoder of one network with a decoder of another network), a new regularizer to impose desirable properties during network training, and so forth. Be creative, ***it might be worth it***!

3. Advanced Pitch



Basic Pitch is a state-of-the-art, open source model for automatic music transcription tasks released by Spotify.

The publicly released checkpoints do not appear to be of high quality, so the the goals of this project are the following:

1. Try to get results that are as close as possible to the ones shown in the Basic Pitch paper on at least two different datasets (Slakh2100 and BabySlakh count aass the same!)
2. Fine-tune your implementation on the hummed music of the “MLEnd Hums and Whistles” dataset

Here are some of the available resources that you can use:

- Paper: https://arxiv.org/abs/2203.09893
- Official implementation (TensorFlow): https://github.com/spotify/basic-pitch
- Unofficial porting to PyTorch: https://github.com/gudgud96/basic-pitch-torch
- Datasets
    - Slakh2100: https://zenodo.org/records/4599666
    - BabySlakh: https://zenodo.org/records/4603870
    - MAESTRO: https://magenta.tensorflow.org/datasets/maestro
    - GuitarSet: https://guitarset.weebly.com/
    - MusicNet: https://zenodo.org/records/5120004
    - URMP: https://labsites.rochester.edu/air/projects/URMP.html
    - MLEnd Hums and Whistles: https://www.kaggle.com/datasets/jesusrequena/mlend-hums-and-whistles
    - MedleyDB (v2): https://zenodo.org/records/1649325
      Access on request, usually granted after some days for academic purposes. Send an email to Daniele (no need to add the others), if not.

4. SynthesizeRL

A music synthesizer is an electronic device that creates sounds by generating and manipulating electronic signals. It can mimic traditional musical instruments or produce entirely new sounds. Users can control various aspects of the sound, like pitch, tone, and volume, to create music.

A major part of the literature focuses on trying to come up with the best neural synthesizer, like [1], [2], [3] [4] and DiffMoog [5], but none tried to use Reinforcement Learning-based approaches, which is the main goal of this project.

Considering the very experimental and unexplored nature of this proposal, you can choose whatever starting point you want, including dataset, model and paper, as long as your approach uses Reinforcement Learning.

Available resources:

- A survey on current trends: https://arxiv.org/abs/2201.02490
- DDSP: https://magenta.tensorflow.org/ddsp
- DiffMoog [5]: https://arxiv.org/abs/2401.12570
- Datasets: see here

For Reinforcement Learning-related doubts, use the same instructions provided in the “How to contact us” paragraph, adding Dr. Antonio Pio Ricciardi <ricciardi@di.uniroma1.it> in the loop as well.

5. Heterogeneous Task Vectors

Task Vectors are an emerging research direction, thanks to a very simple yet useful and practical intuition.

A Task Vector is a vector that encodes the difference between neural network parameters optimized for different tasks. Formally: $v_{A->B} = \theta_{B} - \theta_{A}$, where A and B can be any kind of task, like pre-training and fine-tuning or classification of two different datasets.

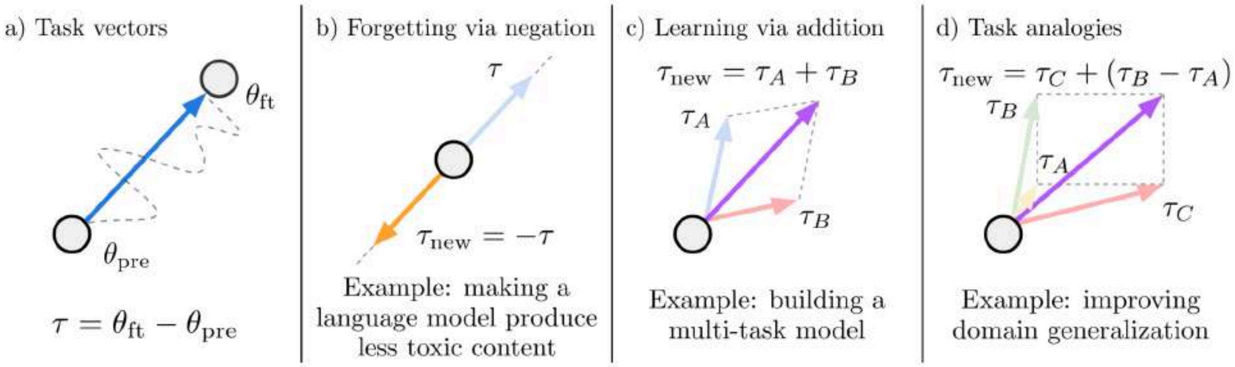

a) Task vectors
θ_{ft}
θ_{pre}
$\tau = \theta_{\text{ft}} - \theta_{\text{pre}}$
b) Forgetting via negation
τ
$\tau_{\text{new}} = -\tau$
Example: making a language model produce less toxic content
c) Learning via addition
$\tau_{\text{new}} = \tau_A + \tau_B$
τ_A
τ_B
Example: building a multi-task model
d) Task analogies
$\tau_{\text{new}} = \tau_C + (\tau_B - \tau_A)$
τ_B
τ_A
τ_C
Example: improving domain generalization

Figure 1: high-level applications of task vectors

This formulation of Task Vectors obviously requires 1:1 matching architectures to work, so the goal of this project is to overcome this limitation:

1. Reproduce one experiment shown in the paper for each application shown in Figure 1 using heterogeneous models sharing some common part (e.g. ResNet-18 and ViT-16-B) and consider only that common part, whenever dealing with Task Vector-related matters.
2. Pick and experiment from one application and try to apply elements of Modular Deep Learning [1] to it.

Available resources:

- Task Vectors original paper: https://arxiv.org/abs/2212.04089
- Other papers related to Task Vectors:
  - https://arxiv.org/abs/2310.15916
  - https://arxiv.org/abs/2405.07813
  - https://arxiv.org/abs/2404.05729
  - https://arxiv.org/abs/2404.03631
- A survey on Modular Deep Learning: https://arxiv.org/abs/2302.11529

6. Audio Task Vectors

Task Vectors are an emerging research direction, thanks to a very simple yet useful and practical intuition.

A Task Vector is a vector that encodes the difference between neural network parameters optimized for different tasks. Formally: $v_{A->B} = \theta_{B} - \theta_{A}$, where A and B can be any kind of task, like pre-training and fine-tuning or classification of two different datasets.

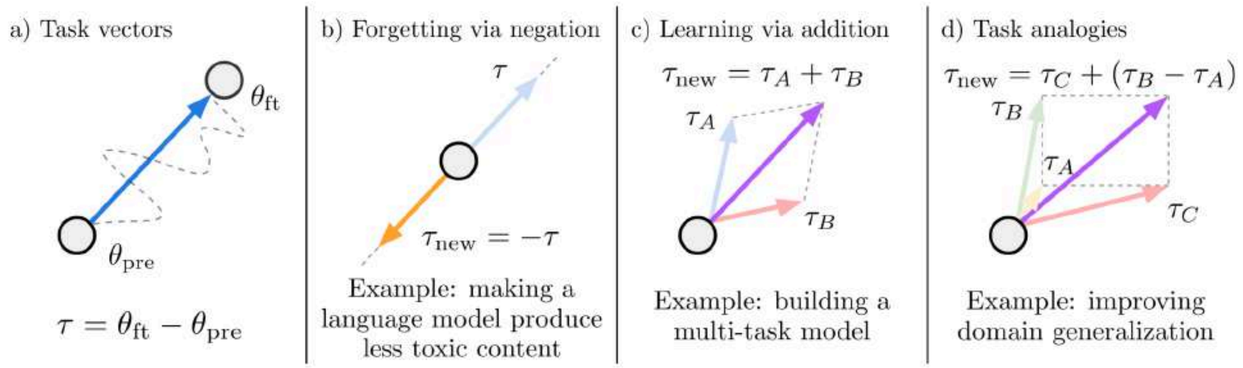

a) Task vectors
θ_{ft}
θ_{pre}
$\tau = \theta_{\text{ft}} - \theta_{\text{pre}}$
b) Forgetting via negation
τ
$\tau_{\text{new}} = -\tau$
Example: making a language model produce less toxic content
c) Learning via addition
$\tau_{\text{new}} = \tau_A + \tau_B$
τ_A
τ_B
Example: building a multi-task model
d) Task analogies
$\tau_{\text{new}} = \tau_C + (\tau_B - \tau_A)$
τ_B
τ_A
τ_C
Example: improving domain generalization

Figure 1: high-level applications of task vectors

The goal of this project is to reproduce applications *b*, *c* and *d* of Figure 1 on audio use-cases. You have maximum freedom in deciding tasks, datasets and models. When dealing with application *c*, please produce at least three task vectors.

Here are some of the available resources that you can use:

- Task Vectors original paper: https://arxiv.org/abs/2212.04089
- Other papers related to Task Vectors:
  - https://arxiv.org/abs/2310.15916
  - https://arxiv.org/abs/2405.07813
  - https://arxiv.org/abs/2404.05729
  - https://arxiv.org/abs/2404.03631
- Datasets: see here

7. Graph Task Vectors

Task Vectors are an emerging research direction, thanks to a very simple yet useful and practical intuition.

A Task Vector is a vector that encodes the difference between neural network parameters optimized for different tasks. Formally: $v_{A->B} = \theta_B - \theta_A$, where A and B can be any kind of task, like pre-training and fine-tuning or classification of two different datasets.

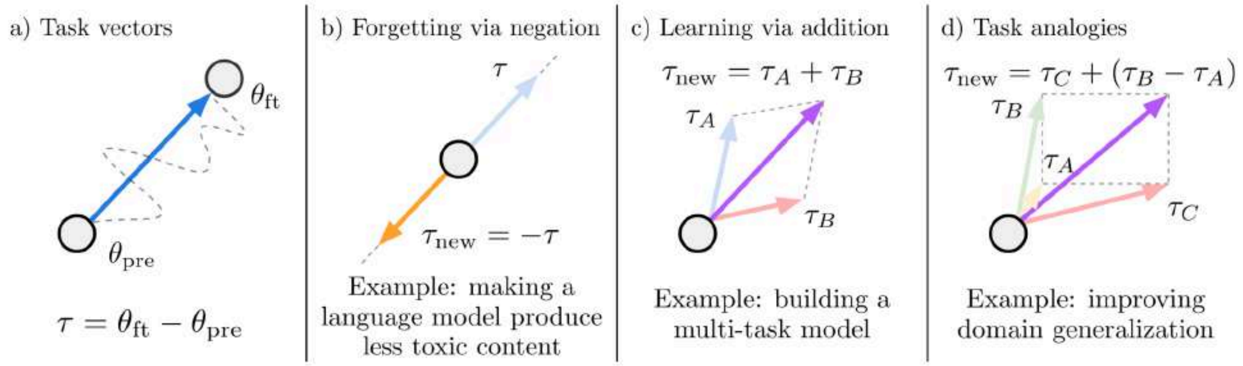

a) Task vectors
θ_{ft}
θ_{pre}
$\tau = \theta_{\mathrm{ft}} - \theta_{\mathrm{pre}}$
b) Forgetting via negation
τ
$\tau_{\mathrm{new}} = -\tau$
Example: making a language model produce less toxic content
c) Learning via addition
$\tau_{\mathrm{new}} = \tau_A + \tau_B$
τ_A
τ_B
Example: building a multi-task model
d) Task analogies
$\tau_{\mathrm{new}} = \tau_C + (\tau_B - \tau_A)$
τ_B
τ_A
τ_C
Example: improving domain generalization

Figure 1: high-level applications of task vectors

The goal of this project is to reproduce applications *b*, *c* and *d* of Figure 1 on graph use-cases. You have maximum freedom in deciding tasks, datasets and models. When dealing with application *c*, please produce at least three task vectors.

Here are some of the available resources that you can use:

- Task Vectors original paper: https://arxiv.org/abs/2212.04089
- Other papers related to Task Vectors:
  - https://arxiv.org/abs/2310.15916
  - https://arxiv.org/abs/2405.07813
  - https://arxiv.org/abs/2404.05729
  - https://arxiv.org/abs/2404.03631

8. Little Tony

Knowledge distillation is a deep learning technique where a smaller, simpler model (student) is trained to replicate the behavior of a larger, more complex model (teacher).

Deep learning models working on music tend to be quite large, so the goal of your project is to apply knowledge distillation to one of the models introduced in the following papers: MSDM [1], LASS [2], LQVAE-separation [3].

Considering the experimental nature of the project, you can choose whatever knowledge distillation technique you want.

Available resources:

- Knowledge distillation:
  - https://arxiv.org/abs/2304.04262
  - https://arxiv.org/abs/2006.05525
  - https://hanlab.mit.edu/courses/2023-fall-65940
- Datasets: see here
- Pre-trained checkpoints: available in the repo of each paper.
  Feel free to contact us, if files get taken down.

9. AWOL - Amusing Wails On Loop

In this interesting paper (https://arxiv.org/abs/2404.03042), the authors were able to show 3D geometry generation ***using language as a bridge***. Can we do the same for ***generating audio*** (speech, individual instruments, music, you name it)?

In this project, you will take the simple pipeline from the AWOL paper linked above and adapt it to the audio setting. According to the authors, the proposed pipeline is very easy to train (down to 5 minutes), requires very few training examples (as little as just one), and generalizes very well out of distribution. Can you achieve this with audio?

You have freedom in choosing the type of data, the external synthesizer / music generation software to use as a parametric model, the final application, and so forth. Chances are that whatever you do will be cool and novel, and it will be fun for the very reason that you are going to generate new sounds! We believe that even noise is fun to hear (so much so that this is even a musical genre), while noisy images are arguably as fun.

Of course, this doesn't mean that a project that doesn't work will get a good grade :)

10. Generating images from a few discrete primitives

The "Imagine" board game challenges players to guess a concept based on the combination of transparent cards.

> *“More than one thousand items from all walks of life can be guessed through the use of 61 transparent cards in Imagine, whether they're placed next to one another or superimposed. Almost everything in the world can be represented by a simplified concept”*

Taking inspiration from this game, we want to make a model capable of generating images based on a few discrete primitives. This peculiarity makes e.g. classic diffusion models unusable. A possible solution involves a multimodal model like CLIP to guide a generation using evolutionary techniques, as described in the work referenced.

In this project, you will:

- Develop a system that uses a combination of a pretrained CLIP-like model and an evolutionary algorithm to generate images like in the "Imagine" board game.
- Build a basic dataset with a few dozen shapes.
- Test the bot's performance qualitatively, and quantitatively against human players in >10 games.

Hashtags: #generative-AI, #multimodal-models, #evolution

References:
"Modern Evolution Strategies for Creativity: Fitting Concrete Images and Abstract Concepts" (https://es-clip.github.io/)
“Imagine board game” (https://boardgamegeek.com/thread/1731037/review-203-by-deskovehry-imagine-ideas-without-lim)

For this project, refer to Dr. Antonio Norelli - Research Associate in CS at the University of Oxford <antonio.norelli@cs.ox.ac.uk>

11. Fine-Tuning a LLM on Italian Dialects

Fine-tuning a recent LLM like Llama-3 or Phi-3, on specific Italian dialects such as Romanesco or Veneto (or your own!) to enhance its ability to understand and generate text in these dialects. If you are interested in this project we will set up a meeting to discuss how to find a proper training corpus.

In this project, you will:

- Assemble a text corpus to train LLMs.
- Fine-tune an LLM on an Italian dialect using LoRA (Low-Rank Adaptation) or other techniques.
- Evaluate the model's perplexity on a held-out test set and compare it with the vanilla LLM.
- Conduct qualitative assessments to analyze the model's performance in generating dialect-specific text.
- Discuss the limitations of a fine tuned LLM with respect to one trained from scratch in mastering a language. E.g. inefficient tokenization.
-

Hashtags: #generative-AI, #LLMs

References:
Examples of finetuning on Alpaca dataset:
(https://github.com/unslothai/unsloth?tab=readme-ov-file)
Example of finetuning of Llama2 in italian (https://arxiv.org/abs/2307.16456v2)

For this project, refer to Dr. Antonio Norelli - Research Associate in CS at the University of Oxford <antonio.norelli@cs.ox.ac.uk>

12. Bot Playing "Dixit" Board Game

"Dixit" is a creative board game where players describe images in imaginative ways. An agent capable of playing "Dixit" may be created by combining a CLIP-like model for image understanding with a large language model (LLM) for generating and interpreting creative descriptions. We want an agent capable of playing both roles in Dixit: who guesses and who gives the hint.

In this project, you will:

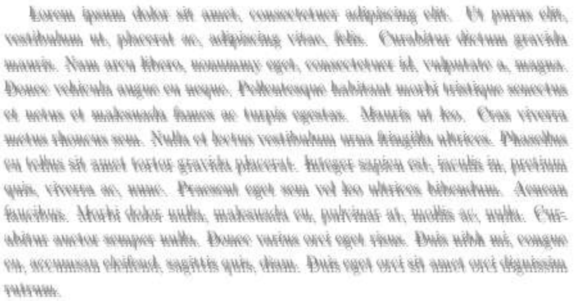
- Develop a bot to play Dixit, based on a pre-trained CLIP-like model to interpret images and a LLM to generate descriptions (GPT APIs are fine).
- Conduct experiments to compare the bot's performance against GPT-4o multimodal and humans in >10 games.

Hashtags: #generative-AI, #multimodal-models

References:
"Creative Captioning: An AI Grand Challenge Based on the Dixit Board Game" (https://arxiv.org/pdf/2010.00048)
Dixit images: (https://github.com/jminuscula/dixit-online/tree/master/cards)

For this project, refer to Dr. Antonio Norelli - Research Associate in CS at the University of Oxford <antonio.norelli@cs.ox.ac.uk>

13. Lossy Text Compressor



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

In this project you will explore the idea of lossy compression for text, a concept usually associated only to images, video or audio. We define lossy text compression as reducing the size of text data while maintaining its core meaning and style (we are not talking about summarization). The idea we would like to bring on the table involves reconstructing text from embeddings –as described in the paper referenced– and pushing it forward through further compression of the embeddings, e.g. through quantization.

In this project, you will:

- Use a pretrained text-embedding model and reconstruct text from its embeddings.
- You will experiment with different quantization levels to analyze the reconstruction quality.
- Choose and discuss proper metrics to compare reconstruction quality.

Hashtags: #compression, #representation-learning

Reference: "On Lossy Text Compression"
(https://academic.oup.com/comjnl/article/37/2/83/491565)
Paper on text reconstruction from embeddings
(https://twitter.com/jxmnop/status/1712562908133999069)
GPT as a blurry jpeg of the web
(https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web)

For this project, refer to Dr. Antonio Norelli - Research Associate in CS at the University of Oxford <antonio.norelli@cs.ox.ac.uk>

14. Coordination of LLM agents in Just One

"Just One" is a cooperative board game where players give single-word clues to help others guess a mystery word. This project aims to create a multi-agent system where different bots collaborate to play "Just One," each one using a LLM to generate and interpret clues. This is a toy problem that aligns with recent and promising research on LLM agents.

In this project, you will:

- Develop collaborative bots based on LLMs to play Just One. Your bots should be able to generate single-word clues and guess the mystery word depending on their role, following the rules of Just One.
- Compare the bots' performance with human game statistics from Board Game Arena for Just One (BGA).

Hashtags: #generative-AI, #LLM-agents

References:
LLMs agents: (https://x.com/AndrewYNg/status/1780991671855161506)
Just one description: (https://boardgamegeek.com/thread/2348677/just-one-a-detailed-review)

For this project, refer to Dr. Antonio Norelli - Research Associate in CS at the University of Oxford <antonio.norelli@cs.ox.ac.uk>

15. Perturbing Safety Alignment Weights

Large Language Models (LLMs) training typically involves two main steps:

1. Pre-training: The model is trained to predict the next word in a sentence using a self-supervised task on a large corpora, which includes a significant portion of the internet and sometimes books.

2. Alignment: Using techniques like Reinforcement Learning from Human Feedback (RLHF)[1], particularly Proximal Policy Optimization (PPO)[2], LLMs are aligned with human preferences to act as helpful and honest assistants.

With the increasing use of LLMs in consumer applications, concerns about their safety and potential harmfulness have been raised[3]. The process of mitigating these concerns is known as "detoxing"[4]. This can be achieved by:

- Data filtering techniques applied to the pre-training data.
- Human-annotated data used during the alignment step.

Recent studies have shown that alignment can be fragile. One study[5] found that alignment can be undone with a few fine-tuning steps, while another[6] identified specific neurons in LLMs that may be responsible for social biases.

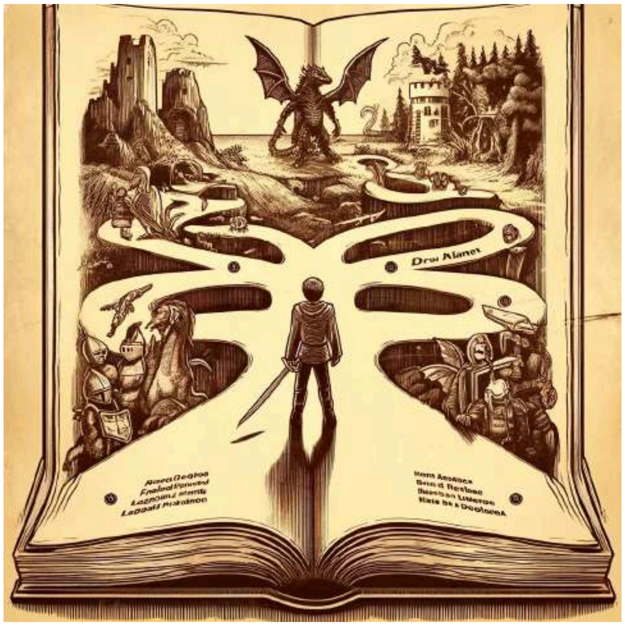
Topics to address:

1. Can we identify model weights responsible for LLM alignment?
2. How sensitive are alignment weights to perturbations?

References

1. Ziegler et al. “Fine-Tuning Language Models from Human Preferences”
2. Schulman et al. “Proximal Policy Optimization”
3. Huang et al. “A Survey of Safety and Trustworthiness of Large Language Models through the Lens of Verification and Validation”
4. Welbl et al. “Challenges in Detoxifying Language Models”
5. Qi et al. “Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To!”
6. Liu et al. “The Devil is in the Neurons: Interpreting and Mitigating Social Biases in Language Models”

16. Choose your own adventure



If none of the above projects catches your interest, that’s totally fine!

We are open to suggestions, especially if centered on the topics proposed in all other tracks.

Feel free to email us using the subject **[DLAI 2023/24] custom project proposal**.

Note: We will not accept every project proposal; in particular, the proposed project ***can not be recycled from other courses***, and it must be related with deep learning and applied AI. If you don’t get our explicit approval on your proposal, it won’t be considered valid for the exam.