EMAT31530: Introduction to Artificial Intelligence Project Information 2023/2024

Laurence Aitchison (laurence.aitchison@bristol.ac.uk)

This is the informational guide for your project. Please consult this in detail, and if you have any other questions please don't hesitate to contact me for clarity.

CONTENTS

1 INTRODUCTION

50% of your grade for this unit will come from a group project. Your group project objective will be to build a system that uses AI to solve a task that is defined by your group. You will then write up an 6 page report, in the style of a research paper, detailing your project.

We will give you the freedom to come up with and execute your own idea as a group, based on what you learned in TB1 in the unit, and be there to support and guide you. If you are having trouble thinking of ideas, you can think whether there are there any problems in the world that you care about that might benefit from AI? Is there something you really enjoy doing that would be fun to involve AI in? A range of projects from previous years are available on Blackboard for inspiration. Each TA on the unit will be conducting research into AI, so if you really can't think of your own task, they will probably have some ideas. The methods you chose to solve the task should draw on those from this unit, but not be limited to them. The top marks go to projects that extend and build on them.

1.1 Types of Projects

There are 4 main types of projects:

- Applications: You base your project on an application that is interesting to your group. You will explore how best to solve this problem using AI.
- Benchmarking: You take a relatively standard task (e.g. classifying CIFAR-10), and carefully examine the performance of a range of a range of pre-existing methods. Some of these methods can be standard methods that appear in the course. But you should include at least one less standard method (i.e. something that already exists, but doesn't appear in the course).
- Algorithms: You develop a modified AI method. Historically, these modified methods have typically been motivated by adapting pre-existing AI methods to an unusual task.
- Replication: Take a pre-existing paper, and re-implement something from that paper.

Projects may consist of a mixture of the above types. For example, in the process of attempting to develop an AI system for an interesting application, you may do a little benchmarking to find which algorithm is best. This is typically a good thing.

The end result should be a research style paper, 6 pages (double column) in length, as well as the system itself, which will typically be code.

1.2 Meetings

Meetings between your team and advisor should each week. However, this should not be the only time you are working on the project. You should be working on it outside of the two hour scheduled on-campus slot, and be using the scheduled slot to discuss progress with your advisor, as well as group work.

In each scheduled session a group member will be required to present a ten minute overview of progress to a TA in the form of a presentation. This should specify which group members have been involved.

2 MAIN ELEMENTS

In this section I will outline the main elements for each project. This is not exhaustive by any means, but it should be considered a guide of what is needed for a typically successful project.

2.1 Project definition

You should have a clear problem you want to try and solve. You should also identify ways in which you can try to solve it.

At the most high level, you should define what the inputs to the system are, what needs to happen with them, and what the output will be.

In order to do well, you need to make sure that the scope of your project is well defined. You want to do something interesting, but it should also be achievable given the time and resources you have.

A very important part of your project definition is the evaluation. Without this, it will be very difficult to measure how well your system works. **In the past some groups have struggled because they did not have a clear idea how to to evaluate their system**. Two general examples of how to evaluate your system:

- For many projects evaluation may require collecting a dataset on which to test your system. Evaluation here may consist of metrics like accuracy, running time, memory usage, and how this compares to other approaches.

More generally, you want to be able to produce some meaningful number that will convince others that your system is working as intended.

2.2 Desiderata

I particularly encourage projects that meet one of two categories:

- It is a fun problem. You can define 'fun' and try to persuade us that it is.
- It is a project about how to use AI for social good. This is very general, but consider if you could successfully build this AI, would the world be a better place?

You are free to ignore this topic advice, and doing so will not affect your grade.

2.3 Actually Building Something

The most important thing to keep in mind that this is a project for a unit on 'artificial intelligence' and therefore this should be the focus of the system.

2.3.1 Language

Given we are building some sort of AI system, we are typically going to need to program a neural network of some description. This neural network must be coded in PyTorch + Python, to make sure that everyone is in a position to contribute to the code. If your project involves building a user interface this may naturally involve HTML, JavaScript and so on. This is also fine.

2.3.2 Version Control

I strongly encourage you to make use of version control (i.e. git / GitHub). Version control systems are software which store and track changes (e.g., Python files) over time. It stores both the current state of the repository and the history. There are numerous reasons for doing so, namely, almost all software is developed this way, and, for this unit, it will record evidence of your contribution to the project over time.

2.3.3 Report collaboration

For collaborating on writing the report, I would recommend Overleaf. The template is provided on Overleaf [1].

2.3.4 The nuts and bolts

Actually building something can involve many different tasks, so it is not possible to enumerate them all. However, most projects will involve the following three tasks

1. Data collection or acquiring a suitable existing dataset.

[1] https://www.overleaf.com/latex/templates/ieee-bare-demo-template-for-conferences/ypypvwjmvtdf

2. Implementing and extending an algorithm.
3. Building some sort of way of evaluating your system, and evaluating a few different versions of the system.

Other tasks may be building a user interface, a simulator, visualizations, to list a few.

Overall, projects vary, so some projects will involve far more of one task, and little of the other tasks. For instance, you could create your own dataset. As creating your own dataset is alot of work, you would then have to do less work on the other aspects of the project (e.g. benchmarking different neural nets). Alternatively, if you're using e.g. CIFAR-10, then you'd have to more work on other aspects of the project. Many datasets can found online, e.g., on built in to PyTorch[2], Kaggle [3], Hugging Face [4], Google Dataset Search [5] as well as academic papers. You may also use pre-existing models (e.g. those in Hugging Face for text [6] or e.g. this repo[7]) for images. Since using existing models/datasets removes a bunch of work, then we expect that if you are using an existing dataset you will do more work on other aspects of the project (2+3).

Importantly, the taught material in TB1 forms the basis for the project. But it is quite possible that good projects will involve elements that you've had to look up and learn for yourselves!

2.4 Approach

An important aspect to consider in your proposal, but also throughout the project, is the approach to your task. You should identify and understand the challenges of the task you are undertaking. Whatever approach(es) you take will have pros and cons, you should be aware of these, and consider and discuss them. Further, you should be aware of the pros and cons, trade-offs etc. specific to your problem.

2.5 Baselines

During this phase it is also important to consider baselines. That is, with what can you compare what you built with in order to measure the success. This is a key part of evaluating your system. Sometimes a baseline may be always predicting the most frequent label in your dataset, or another may be what would the performance of acting randomly be. Alternatively, a baseline might be a very simple machine learning system, such as linear regression/classification, or the first simplest network you tried.

If you are using an existing dataset, it is likely that there are other methods and approaches that have been applied to this dataset. You can use these as a means of evaluating your own. If you have collected your own dataset, then you may want to look at other methods and approaches and compare with these too.

2.6 You can still do a good project, even if the system "doesn't work"

This is a coursework, not a research project. Therefore, it doesn't matter to me if the system "works" or doesn't "work". What matters is that you've done careful work to explore the system. So capture code + results even for models that don't work! If your model doesn't work, I want to see is that you've thought carefully about the results, formulated hypotheses for why the system might not be working, and tried to fix those issues. For instance, if your system isn't working, one hypothesis might be that you haven't trained the system for long enough. You could test that by plotting accuracy/loss vs training epochs. So ... tell me all that! And report how the results shaped your future approach! Additionally:

[2] https://pytorch.org/vision/stable/datasets.html#built-in-datasets
[3] https://www.kaggle.com/datasets
[4] https://huggingface.co/docs/datasets/index
[5] https://datasetsearch.research.google.com/
[6] https://huggingface.co/docs/datasets/index
[7] https://github.com/kuangliu/pytorch-cifar

- Trying out stuff that ultimately didn't work can also form an important part of your individual contribution, even if it doesn't make it to the final report. Include details on this in your one-page document detailing your contribution (see Sec. 3.
- The above is also good advice for final year projects!

2.7 Literature review

This is also an important component of any successful project. It is very unlikely that your idea exists entirely within a vacuum. Other people may have attempted to solve the same problem, or build a similar system. You should be aware of those, including the strengths, weaknesses and ways in which they differ. You should provide a detailed examination of the relevant literature, and not merely summarise it. Academic search engines such as Google Scholar tend to do well at surfacing relevant papers given appropriate search terms, as do normal search engines.

However, as this is a coursework rather than a final-year project or a research project, your project is not expected to be "novel" against previous work in the literature (e.g. I don't really think the ideas in the example projects are really "novel"). Your work will be judged on the effort and quality of your development work, rather than on the "novelty" of the idea.

2.8 Experimentation and Analysis

This will likely be a major part of your project. This will be concerned with designing and implementing experiments to evaluate your system. In order to do this you will need a some hypotheses to test. You should identify some in your project definition, and you may have some more by the time you reach this stage. In order to understand the experimentation you will have to do significant analysis. You should analyse properties of your approach, the experiment, and discuss them in adequate detail. Where did your approach work well, where did it perform poorly, and why this is the case should all be investigated. Naturally, you should reach some sort of conclusion in this section which should be discussed.

3 GRADING INFORMATION

Your final report should be between 6 pages using the following template (`https://www.overleaf.com/latex/templates/ieee-bare-demo-template-for-conferences/ypypvwjmvtdf`) and be accompanied by a URL to the system itself on GitHub. It should typically contain a completed form of all of the elements described in Section 2. **This will be worth 50% of your grade.** In summary, the final report should typically have the following sections (not necessarily in this order):

- Abstract (200-250 words usually)
- Introduction
- Literature review
- Method
- Experimentation
- Discussion
- Conclusion
- Future work

Some projects will naturally not fit into this exact structure. You shouldn't need something radically different, but if you want to tweak the headings, check with your advisor. This report should be written in the style of a scientific publication. You can have a look at a few examples of projects submitted last year.

Each student will also have to produce a one page document detailing the major elements of your role in the project, and pointing to evidence (e.g. GitHub records) if available. Each student must contribute an adequate level of meaningful code to the project. Your team will, at the end, distribute points to each member of the team, representing how much they believe you contributed to the project. **Importantly: the final weighting of your mark is decided by me (Laurence) taking a wholistic view of all the information related to the project. There is no simplistic "algorithm" that takes your suggested** The due date for the project will be March 20th 2023, at 1pm.

4 FREQUENTLY ASKED QUESTIONS

4.1 Will I be assessed on the code?

1. **For the project mark:** The code will not be directly assessed for the project. More code does not necessarily mean more marks. Code, however, can be used as a proxy for other elements the report is assessed on, particularly when there is uncertainty from the report on how much work was done.

2. **For the individual mark:** Sometimes not all members of a group will receive the project mark. Such an event occurs when peer assessment raises the issue that not all members contributed equally. In these cases, the code may be used to assess an individuals contribution to the project.

4.2 What will be assessed then?

The project will be primarily assessed on the content of the report. Have a look at the proposed section headings in Section 3.2. The bulk of the project time will be spent in the method, experimentation and analysis sections. Here, you should be iterating on your methods, analysing the results, investigating improvements and then evaluating the new results. This is typically what is done in both academic research, and when working on AI and machine learning in industry. For nearly all (if not all) projects you will typically follow the 'Machine Learning Workflow' as described in the lectures, where you build a system, evaluate the results, and tweak your system based on this. We want to see you demonstrate your knowledge and understanding of AI by taking what is learned in this unit, applying it to difficult problems, understand how and why it works (or doesn't work) and then iteratively improving upon it, either with other ideas from the unit, or extensions of those from your own research.

4.3 How do I use GitHub?

Please carefully read, and follow, the Introduction to Git and GitHub[8] tutorial on Blackboard under 'Resource lists'. If you follow the steps on how to use GitHub desktop, you will learn the basics. This is what you need to undertake this project.

4.4 Can you summarise what the main GitHub steps are?

1. Please read the tutorial mentioned above... but to summarise...

2. The first step is to 'clone' the repository from GitHub.com onto your local computer. This will store the all the files in a folder or directory on your computer. You should now store your work within the same directory. This is kind of like 'dropbox' but instead of syncing being automatic, you have to manually carry out some of the following steps in order to 'sync'.

3. When you create new files or folders, or make changes to existing files or folders, and you want to store them in Git, you must first *'commit'* them. Committing stores a version of the file in your local repository. You can think of this as saving your file to Git.

[8] `https://www.ole.bris.ac.uk/bbcswebdav/pid-6971281-dt-content-rid-31823297_2/xid-31823297_2`

4. However, this is saved only on your local Git repository. You will want to share this with the rest of your team. To do this, you will want to *'push'* your changes to GitHub.com
5. Finally, periodically you will want to do a git *pull* or *fetch*. This will pull down and changes from the GitHub.com repository to the folder on your local computer, in other words, it will pull down changes made by others in your team. Git will try to merge the changes automatically.

4.5 Do you have any advice on using Git/GitHub?

My advice is to commit your changes often, push your changes to the GitHub.com server often, and fetch changes from your team often. This may seem annoying at first, but it is a good way of learning to use GitHub and it will be worth it when more than one person is working on the project at any one time.

4.6 Do you have any example project reports from previous years?

Yes they can be found on Blackboard.

4.7 Must the report be in LaTeX?

No, but it is preferred. Overleaf is highly recommended as it makes collaboration very easy.

4.8 Does the page limit include references?

No, the page limit excludes references. Reference often! If you're using Latex (Overleaf), referencing is made easy using BibTex (see, e.g., `https://guides.rider.edu/overleaf/citations`).

4.9 How do we say how much each person contributed?

Towards the submission deadline, I will be sending out a form where everyone must specify, as a percentage, how much each member of the team contributed.

4.10 Can I include a demo or video?

Yes, feel free to a URL include an online demo or video of your system. You will not be marked on this however, and therefore your paper must, without a demo or video, fully explain your work.

4.11 Can you provide examples of what is expected for a reasonable project?

Previous examples of good projects can be found here on the Blackboard unit page.

5 SUMMARY OF MARKING CRITERIA

As a summary of the points above, around the marking criteria:

- A good project will have a well-defined goal.
- A good project will often start with a simple proof-of-principle, and move on to something more complex.
- Your project will necessarily involve a number of choices. For instance: why did you use this particular dataset? why did you choose to reduce your images resolution? why did you choose a particular base network? why did you choose a particular evaluation strategy? how did you deal with a lack of compute? A good project will have a thorough discussion of some (not all) of these choices.
- A good project will have some numerical results (e.g. performance of the network on a dataset), and discuss of the implications of those numerical results (e.g. do the results indicate the system is performing above chance? is the system performing well relative to some sensible baseline (e.g. a human)? might the results be spurious? might you have a bug? if so, where might the bug be?)

- This is a coursework: it doesn't really matter if the system "works" or doesn't "work". What matters is that you've documented your choices, and carefully thought-through and discussed the implications of your numerical results.
- The project is primarily marked on the report, though I may use the code as a proxy for other elements, e.g. when there is uncertainty about how much works was done.
- If relevant to the application, a good project will discuss the social and ethical issues (e.g. would be relevant for facial recognition systems, but not for benchmarking on CIFAR-10).
- Group size will be taken into account as part of the marking process (though there isn't a rigid "formula").