CS3244 Project Quick Start Guide

This quick start guide is supposed to help jumpstart groups without particular directions as to how to go about starting to define a project and learn appropriate resources. You can take this as a short complementary guide to the official slides; this guide was primarily put together by your seniors and alumni from this course as they were as lost as you are probably feeling now.

What is the goal of this project?

The goal of this project is to assist you, as an ML student, to familiarise yourself with the workings of an ML project: identifying a topic of interest, searching for relevant datasets, and conducting ML experiments in your chosen field.

ML is a field that is both highly theoretical and practical at once. It is our hope that while you will gain the essential ML knowledge through our weekly lectures and tutorials, you will put this knowledge to use through this project.

We recommend that you define an easy, very limited project to *complete* around Week 8 so that you can have a clear project finished and extendable from. It's generally easy to make a small project bigger (e.g., by removing some limiting assumptions) than to make a big project smaller. This is the software engineering agile methodology in a project form.

Expectations for each skill level

We understand that many of you enter this course with different degrees of expertise with respect to ML. Hence, we have outlined some expectations we have for students of different skill levels. Note that these expectations are not rules set in place – they are more for you to use as a benchmark to pace yourself throughout the course of the semester. Above all, focus on learning and stretching yourself. With relatively large team sizes of 6, you have the opportunity to study multiple problems or models: subgroups within the group can study different projects and then reconvene towards the end to take stock and find linkages between the different substudies for overall conclusions.

For students interested in deep learning, please note that these projects take significant amounts of data and compute to work well. Thus the difficulty of these projects is accordingly higher. Not only do these projects require a bigger initial investment from you (as you will have to self-learn much of the work before formally learning it in our course), but also the training and testing will be heavier (takes more time, and if you start late on your project, you will have to compete more with others).

Beginner (little to no experience in ML)

Orientate yourself with the basics of an ML project, including how to set up the necessary tools and considerations when faced with a machine learning project. In addition, learn how to apply the different models taught in class (such as SVM and NN) and compare their performance. Evaluate why certain models work better than others for your project: the key is to not blindly apply libraries, but also attempt to understand why things work a certain way.

Focus on datasets where the data is clean so that you can focus on the ML algorithms instead of collecting and cleaning data. This will take a portion of the difficulties with doing ML out of the scope of your project. We have tried to vet datasets that we feel are largely clean and where our TA and staff have sufficient experience to help guide you.

Intermediate/Advanced

Aim to define your own problem statement, since thinking of how to apply ML to existing problems is a highly valuable skill. Of course, if you feel strongly about any existing problem statement, please feel free to proceed with that problem.

You are also encouraged to source for and collect your own data to supplement the existing vetted datasets: this is more realistic a scenario to how ML is applied in industries, whereby data is unclean and a fair amount of preprocessing has to be done on the data before ML algorithms can be applied on them.

Basic Machine Learning Resources

A list of popular machine learning resources is included to help you get your feet wet. While you may wonder the purpose of self-learning machine learning when you are already taking this module during the semester, the honest truth is that ML encompasses such a wide domain that the teaching staff is unable to cover all the basics within the semester. Our staff have curated the resources and material that we feel is most helpful for a practical introduction. We can only hope to cover as much as we can, and it will be immensely valuable for you to learn other topics on your own.

In addition, going through these courses will further reinforce your familiarity with actual implementation of ML algorithms, a crucial skill for this module.

1. Andrew Ng's Machine Learning course on Coursera: This course is widely benchmarked against for good reasons – it covers the basics extremely well, and there are also plenty of coding exercises to get a feel of the algorithms. However, less emphasis is placed on using popular libraries such as Keras. Hence, you may not be proficient with using those libraries even after finishing this course.

2. Kaggle Learn: This section on Kaggle covers different skills, ranging from data visualisation to writing deep learning code. While they are less theoretical than Andrew Ng's Machine Learning course, their focus on using external libraries means that you are able to acquaint yourself to these technologies fairly quickly.

3. Dive into Deep Learning: This course and set of notebooks uses the MXNet (vs framework. The book has accompanying Jupyter Notebooks (as does our class), and taught in UC Berkeley. MxNet is different from PyTorch, but similar in syntax and has demonstrated some performance improvements. MxNet is also adopted by Apache and is more devoted to open source.

Setting up Machine Learning Technologies

Software

Setting up a machine learning environment

Anaconda is a data science platform that allows you to set up machine learning environments. Using Anaconda is highly recommended as machine learning libraries often have dependencies: one version of a library is reliant on another library being of a certain version. Creating a specific machine learning environment helps your project by ensuring that you have a compatible set of machine learning libraries installed- if you wish to upgrade a library, you can always create another environment without violating the existing dependencies in your project.

Popular machine learning libraries

We list popular machine libraries, along with a link to their basic tutorials below.

1. Traditional ML: SK Learn
   SK Learn is a useful library for tools like creating a train–test split, and running other algorithms not taught in our course; you can cover these on your own self-study time. Our previous iterations of CS3244 have more lecture topics although the videos for them are not particularly good. You can find the playlist from AY 20/21 Sem 1 on YouTube here.

2. Deep Learning:
   a. PyTorch: This library is probably the one that is most useful for learning deep learning from a pedagogical point of view. Its abstractions make the most sense for connection with the literature. It is a good library also for doing research, and you'll find most research teams investigating deep learning at NUS (rather than just using it) via PyTorch.

b. Keras: Keras does not have an authoritative beginner's guide, but this library is easy to use and many other tutorials have clear code that utilise Keras library.
c. Tensorflow: This is the best framework to learn if you think you'll want to use deep learning in your eventual industry job. It is a highly optimised learning for production settings and can deploy and scale quite well for both low-end use (on a mobile phone) as well as high-end use (on large clusters of data center computers). Tensorflow also has very good integration with analytics for ML; in particular the Tensorboard dashboard, whose implementation has been mirrored in other frameworks.
d. MxNet: The framework adopted by Apache and used by the Dive in Deep Learning textbook/Jupyter notebooks. MxNet also works well in cloud frameworks such as AWS.
e. fastai: Fastai is a deep learning library built on top of PyTorch with the goal of making AI easy and accessible to all.. It is beginner friendly and provides high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains for vision, text and tabular data.

3. Reinforcement Learning: OpenAI Gym

   We do not recommend that you take a RL topic as your core project, but you may want to do this as part of an extension of your core.

   This is the current favorite framework for deep reinforcement learning. Several of the TAs have used this framework for projects before and may be able to advise. It has efficient implementations of many RL frameworks.

Hardware

Training on your computer is often infeasible: the code may run too slowly and your computer may overheat. Plus you probably don't want to lock up your computer for a long time and make it slow even to do simple tasks. Thankfully, there are other cloud resources available for machine learning code.

Tembusu Cluster

Servers are available on the Tembusu cluster for NUS students to run their machine learning code. Instructions on how to access these servers can be accessed via this link. Do note that this cluster tends to be very popular (i.e. you have to compete with other users for CPU/GPU) towards the later part of the semester; hence you are encouraged to start running your code early if you plan to use this service.

For this semester (AY 21/22 Sem 1), we have 10 dedicated compute nodes that you can access to do your project work in your groups (`xgpf0-9` from 4 Oct to 12 Nov). We will also have Google Cloud Compute credits for you ready soon.

National Supercomputing Centre

NSCC is Singapore's official supercomputing facility. You are able to create an account to utilise this facility via this link. There have been recent changes to make NSCC more suitable for deep learning projects, but previously staff have found installing new libraries troublesome or impossible to do with the permissions given to ordinary users. If you want to use this resource, check the configuration and NSCC documentation to ensure that you will be able to run the appropriate frameworks and resources to complete all aspects of your project before committing to this resources.

Google Colab

Google Colab is a free Jupyter notebook environment where you can also run your machine learning code. You can also use up to 12 hours of free GPU per usage. However, if you are using modules that are not installed in the default environment, you will have to reinstall them when you connect to a new VM. In addition, Colab can be slow when you are loading a large amount of data, something common in machine learning. These traits make Google Colab perfect for prototyping a simpler model before scaling it up and training the full model.

Paid utilities

Google (GCE), Amazon Web Services, Microsoft (Azure) and Alibaba (Alicloud) are other popular commercial services that allow you to host your own code. They provide first time users with complimentary credits. Many of these multinationals also have libraries and GUIs to run ML algorithms without all of the tuning knobs ("above the hood") that we explicitly want you to learn as part of this introductory ML course. As we need you to understand the mechanics of what is happening "below the hood", we don't advise that you use these resources for the foundation of your project, but you can certainly try these out to get a sense of what is possible. These include Google Cloud Machine Learning Engine, Amazon Sagemaker, among others. These products are good for data scientists who just want the black box ML algorithm to work, but are largely unsuitable for our class because they obscure the necessary understanding of the fundamental ML principles. We plan to release Google Compute Engine credits of $50 per student, hopefully around Recess Week, so that you have a try on Google's Cloud platform if you choose.

Conclusion

We hope that this guide helps to clarify some of the doubts you may have regarding how to go about executing a machine learning project. If you have any questions, please feel free to ask them on the Coursemology forum, and approach your tutorial leader or come see the staff during the help sessions.