Team Project

CSCI 4850/5850 - Neural Networks

Associated due dates and requirements for each component are listed below.

Overview

For the remainder of the semester, teams will develop a research-based project that applies neural network (a.k.a. deep learning) technologies to at least one engineering domain of interest or empirically explores the capabilities of these technologies. As a team, you will need to formulate the overall project idea and follow through the implementation of the project so that it meets the requirements below. All projects will be released to the public under an open-source license (MIT or GPLv3) which makes them amenable to future development as well as sharing with potential employers and other interested parties. The repositories will remain public throughout development of the project, but no developers/persons outside of the course will be allowed write access to the repositories. Pull requests from outside developers will be ignored until the completion of the course projects at the end of the semester. Please *keep these things in mind* while you are developing your projects. While it is unlikely that anyone outside of the course will become interested in the course repositories until after all projects are finished, they are still publicly visible and reflect the level of professionalism that you exhibit during their development. Once the projects are finished at the end of the semester, you can (and should) fork the repositories to your own accounts for continued development and self-education in the field. Finally, since you will be committing information to a public repository, you are at-risk of *publicly plagiarizing material from other projects*. While inspiration from existing projects and research is allowed (in fact, highly encouraged), you must properly cite all such materials. If you have **any** questions about how to cite **any** material, please ask the course instructor for guidance: infractions may result in a grade of zero for the project assignment.

Project Components and Overall Point Allocation

Team projects will consist of several interacting parts, all of which will require team **communication**, **coordination**, and **effort** for a total of **400 points** for the entire project:

1. Project **proposal** development - (**12**) - due: **Mar. 8 @ 11:00pm**
   The initial part of this assignment will consist of preparing a proposal for your team project: details are listed below. The main thing to keep in mind during the proposal process is that is it **not** binding. You will be allowed to make changes to your project as the semester progresses *if needed*. Changes to the overall scope and/or specific nature of a project will require modification of the **proposal document** over time to match the evolution of the project.

2. Project **milestone** management - (**48**) - due: **Mar. 15, Mar. 29, Apr. 5, Apr. 12, Apr. 19, Apr. 26 @ 11:00pm**
   In order to assist project development, team-self-assessment, and team-self-reflection, each team will be required to compile a weekly milestone report indicating the current status of the project, work completed since the last milestone, and anticpated work that will be completed by the next milestone. Details on how to construct these short reports are shown below.

3. Project **code** development - (**40**) - due: **milestone driven**
   Your core project idea will undoubtedly require the development of Python3 code (scripts and/or notebooks), other code (anything not related to neural network building/training could use other languages/tools), curation of project data, test case development, etc. Most code developed for the project will be part of this portion of the assignment, and should be focused on solving the chosen project aim(s).

4. Project **paper** development - (**75**) - due: **Apr. 23 @ 11:00pm**
   The project **paper** represents academic documentation of the completed results and will be due *before* the project demos and presentations. The paper will focus on describing the backgroud of the project and technologies employed, details of the code and analysis performed on the project, explanations of figures and data tables related to the project results, and final lessons learned from the project. It is not focused on the project demo, but will probably cite one or more of the github project repositories for reference. Additionally, looking up relevant background (textbook chapters and research articles) for filling in background material for the project is *expected*.

5. Project paper **peer-reviews** - (**30**) - due: **Apr. 28 @ 11:00pm**
   After the project papers have been submitted, you will be asked to read a paper from another team and submit a short, formal review of the paper. Reviews will be single-blind, and returned to the reviewees once completed.

6. Project **demo** development - (**60**) - due: **May 3. @ 9:30am**
   Even after a project has been fully implemented, results have been computed, and something useful has been learned, it needs a proper non-academic presentation for the outside world. A project demo utilizes the core code developed to produce an interactive tutorial (could consist of notebooks, scripts, instructions, graphics, etc.) that allows those unfamiliar with neural networks to see a general introduction to and results of your project. Demos will not be due until the day of finals and will be assumed to play a key role in the project presentation.

7. Project **presentation** development - (**35**) due: **May 3. @ 9:30am**
   Project presentations will be 11-13 minutes long for each team and utilize a computer and projector (i.e. in the classroom). Slides and other media are encouraged. Follow-up question/answer time will be contained to 2 minutes post-presentation. Other details related to the presentation material are listed below.

8. Team member contributions **assessment** - (**60**) due: **May 3. @ 11:00pm**
   Contribitions made by team members will be analyzed in light of documented team member assessements. This means that each team member will participate in the assessment process by completing an assessment form for each of the other team members. Well-constructed and accurate assessment of other team member contributions is required, and will be graded on a **individial** basis. The focus of this part of the project is to ascertain contributions made by other team members to the project

9. Team member contributions **allocation** - (**40**) due: **May 3. @ 11:00pm**
   Points for team member contributions will be allocated on an **individual** basis and will be based directly on team assessments.

Each of the above parts of the assignment is described in more detail below. NOTE: This document is a work-in-progress, and additional details on the portions of the team project may be adjusted closer to the times that they are expected to begin.

Part 1 - Project Proposal (12 Points)

Due: Mar. 8 @ 11:00pm

Requirements

Your project proposal should consist of at least 500 words (max. 1000) describing the overal goals of your project, the neural network technologies you intend to use, the potential data or data sets you intend to leverage, and the kinds of tests you would like to perform to verify that your final product works as intended. Remember, these are not detailed technical descriptions, but you should provide enough detail to convince a professor (or manager) that you

have an **feasible idea** and that you know how to determine the **success** of your idea in a both a **quatitative and qualitative** manner.

Advice

Constructing a proposal is a good way to start a project since it forces you to think about all of the key components that make up a complete project, and how they will fit together as a whole. It might also bootstrap the process of assigning roles for different team members and individual work assignments. Every team member's input should be collected for this assignment, but it is natural for certain team members to take on different roles. Some people may decide to take the technical lead and develop the core code and tests for the technical direction of the project. Others may decide to take a managerial role in order to provide support in writing up the relevant milestone reports or other project reporting (including this document, the paper, peer-reviews, etc.). Even though you are not required to report these contributions in this proposal, your first milestone report will require that you have have worked out some of these issues so it's also good to think about these issues as a team and decide how you will delegate responsibilities during this part of the project.

Rubric

1. Does your project have a specific or focused aim (or set of aims)? (3 points)
2. Does your proposal exposit the specific role that neural networks will play in the project? (2 points)
3. Does your project have a specific set of data (or data sets) identified? (3 points)
4. Does your proposal describe a verifiable testing protocol? In order words, did you describe a set of specific tests that you could run for collecting data and/or statistics which can discriminate between success or failure at meeting the specific aim of the project in both a quantitative and qualitative manner? (4 points)

Submission

Your project proposal should be constructed in an iPython notebook file (.ipynb) consisting of only markdown cells, should be aptly named (eg. Project Proposal.ipynb), and should be committed to the **project** repository for your team (eg. S21-team0-project) in the top directory of the repository.

Part 2 - Project Milestones (48 Points; 8 points each)

Due: Mar. 15, Mar. 29, Apr. 5, Apr. 12, Apr. 19, Apr. 26 @ 11:00pm

Requirements

Your milestone document should contain a table like the following:

Deliverable	Percent Complete	Estimated Completion Date	Percent Complete by Next Milestone
Code	15%	Apr 14	30%
Paper	0%	Apr 21	15%
Demo	0%	May 1	5%
Presentation	0%	May 3	10%

Peer-reviews have specific deadlines and project milestone updates are recurring, so those deliverables do **not** belong in the table.

You should then write a short response (one or two sentences) for each of the following questions:

1. What deliverable goals established in the last milestone report **were accomplished** to the anticipated percentage?

2. What deliverable goals established in the last milestone report were **were not accomplished** to the anticipated percentage?
3. What are the main deliverable goals to meet before the next milestone report, and who is working on them?

Advice

Communication is critical for crafting a milestone report. Whether you choose to prepare the report as a group, or to delegate to an individual or subset of the group, you will need to communicate with each other to determine the roles and deliverables each team member is going to contribute in order to fill out a successful report. The point of this exercise is accountability and awareness. Setting goals and sticking to them as best a team can will help the team make progress throughout the semester instead of only at the end. This will be important because set-backs are common, and having time to plan and execute an alternative strategy may be needed.

Rubric

1. Does your report have the necessary table, filled with the correct information? (2 points)
2. Does your report answer the questions above clearly and concisely? (3 points)
3. Does each team member have a role to play in helping to meet the goals of the next milestone? (3 points)

Submission

Your project milestone document should be constructed in an iPython notebook file (.ipynb) consisting of only markdown cells, should be aptly named (eg. Project Milestones.ipynb), and should be committed to the **project** repository for your team (eg. S21-team0-project) in the top directory of the repository. You can update this document continually throughout the semester, adding new markdown cells for the next round of reporting.

Part 3 - Project Code (40 Points)

Due: milestone driven

Requirements

Code requirements are generally not very strict, but it is expected that you will utilize Keras/Tensorflow for neural network development and training. Also, focusing on scripting languages like Python/R/bash instead of compiled languages is preferred. The main requirement is to make regular use of the **project** repository (eg. S21-team0-project) for code development. Team members should make regular contributions to the repository for logging progress on the code. Progress will be monitored based on commits to the repository.

Advice

Commit often during code development. Make personal branches and merge any changes in the master branch into your own personal branch before merging your branch back into master. Be careful not to commit large files into your repository. Large data sets, and notebooks with lots of plots and large vector/tabular outputs can bloat the repository unnecessarily. If a notebook is being used for development, try using the "Restart Kernel and Clear All Outputs" option to clear the output section of all cells before committing changes to the notebook into the repository. Try using URLs and file hosting methods for large files instead of adding them to the repository. Large is typically considered something more than a few MB in size. If you feel you **must** store a large file in the repository, be sure that it will **not** need to be updated often. For example, long log files from testing runs should not be added to the repository since they change often and committed changes will slow down push/pull/clone operations on your repository. This is because, unlike other version control systems, a cloned git repository contains -all- updates from the first commit to the last.

Rubric

1. Do team members make regular commits to the repository (at least weekly)? (7 points)
2. Is the repository well-managed in that it doesn't contain a lot of unnecessary file bloat? (3 points)
3. Does the code function for its intended purpose? (30 points)

Submission

Project milestones will be used to inform project progress, and regular commits to the **project** repository should reflect relevant progress towards those goals.

Part 4 - Project Paper (75 Points)

Due: Apr. 23 @ 11:00pm

Requirements

The paper for the project will consist of a write-up with the following four sections:

1. Abstract
2. Introduction and Background
3. Methods
4. Results
5. Discussion and Conclusion

The minimum paper length will be 3 pages formatted using a combination of notebook markdown and LaTeX. The **Introduction and Background** section should focus on developing the overall motivation for the project, reference related work on similar problems or that has used similar approaches, and indicates where your project provides some additional insight into neural network technologies. Be sure to clearly state your key aim(s) at the end of this section. The **Methods** section should focus on the mathemetical, theoretical, and practical execution of your project by introducing and referencing the employed technologies, data sets, statistics, and visualization techniques used in the project and **why** they are being used to address your domain or problem of interest. Algorithmic, mathematical, and step-based descriptions may be presented, but you should generally provide **no code** in the Methods section. The **Results** section should provide the results generated using the employed methods including figures, graphics, data tables, statistics, etc. This section should also contain ample text that walks the reader through the key findings from your study, and uses the previous aids to **support** the main text. Take little for granted in the Results section, so that a reader can follow your logic and understand your results by using text that explains how the various figures, tables, etc. provide evidence for your conclusions. The Discussion and Conclusion section should revisit your **main** insights gained from your project and how these fit into the ideas presented in the Introduction and Background section. Be sure to explain how your project has provided some additional insight into the specific idea mentioned at the end of this section. Also, feel free to provide some examples of new directions and ideas where your would like to see this work unfold in the future. Finally, the **Abstract** should be written last, and should be no more then 250 words that takes the key ideas from each section of the paper and composes them together. That is, you should mirror your abstract after the sections of the paper. You will only be able to write one (or maybe two) sentences that summarizes each section. The most common mistake is to leave out the key results or findings from your project out of the abstract: don't do that!

You will need to cite information properly for this assignment, and follow the paper template provided in class to construct your paper.

Advice

Scientific writing is an art instead of a science. You should think of your paper as taking the reader on a journey through an imaginary realm of digits and data. OK, maybe not as cheesy as that sounds: it's just a metaphor, so don't take it literally. On the other hand, you are not trying to write for art's sake alone. You need to provide motivation, exposition and convincing analysis so that the reader is not left guessing how you accomplished some task and would also agree with you that your results are indeed consistent with your explanations. Being too qualitative can become unconvincing since you will be tempted to rely on intuition or guess-work alone, and the reader will want to call your bluff. Being too quantitative can become too unconvincing since you will be tempted to rely on a numerical value that is inadequately explained or understood, and the reader will want to call your bluff. Be a good story-teller, but remember that this is nonfiction.

Start writing without the template! You don't need to do anything more than start putting words down on paper, and then rearrange them into something worth submitting. Most paragraphs of a paper should be rewritten **at-least** one time to ensure they fit with the other parts of the manuscript. Be careful not to use concepts before you have introduced them in the manuscript. Use bibilography management tools (Zotero, Mendeley, EndNote, etc.) to manage your references: don't write the bibliography by hand.

Rubric

1. Does your Abstract provide a summary of all key paper sections? (8 points)
2. Does your Introduction and Background section contain adequate references to indicate some project significance? (7 points)
3. Are your aims clearly introduced and succinctly stated? (15 points)
4. Are your Methods explained clearly enough that someone without the code could reproduce the results (additional effort only)? (15 points)
5. Are your Results explained and media presented clearly and do they agree with your provided media? (15 points)
6. Does your Discussion and Conclusion section revisit/redress your aim(s) established in the Introduction and Background section? (8 points)
7. Does your Discussion and Conclusion section explain how your results might be used in the future? (7 points)

Submission

Your paper should be constructed in an iPython notebook file (.ipynb) consisting of only markdown cells or LaTeX source (.tex,.bib), should be aptly named (eg. Project Paper.ipynb), should include a compiled version (eg. Project Paper.pdf), and should be committed to the **project** repository for your team (eg. S21-team0-project) in the top directory of the repository.

Part 5 - Project Peer-reviews (30 Points)

Due: Apr. 28 @ 11:00pm

Requirements

You should construct a document that provides a thorough review of a final paper from your peers (I will email the PDF you will be reviewing to your MTMail accounts the day after the papers are due). For this assignment, you should NOT put your name on your review document since all comments will be provided back to the authors anonymously (before grading). Each section of the review should be at least one paragraph in length, and the general steps for constructing a review document are as follows:

1. Include a header with the following information:
    - Team Number: X (the *author* team number)

    - Title: Researching Research for Research's Sake (the *paper* title)
    - Rating: [Accept, Weak Accept, Weak Reject, Reject]
    - Review Confidence: [Low Medium High]
2. Write a summary of any formatting or structural issues with the paper. This section should attempt to assess whether or not the paper does not meet the proper formatting guidelines (i.e. does or does not use the IEEE template properly). The name of this section should be **Format**.
3. Write a summary of your teams' overall impressions with the paper, and any additional comments that you would like to make to the authors. Specifically, you should suggest things that should be done to *strengthen* the paper. These should either *mitigate a weakness* expressed earlier in your review, or *reinforce a current strength* that was somehow overlooked in the paper. The name of this section should be **Summary Statement**.
4. Write a numbered list of proposed revisions under the following section headers:
    - **Major Revisions**,
    - **Minor Revisions**, and
    - **Discretionary Revisions**.

Advice

Your **Rating** is essentially whether you think the paper would be rejected or accepted at a regional machine learning conference. I know most of you will have little or no experience with this, but give it some thought. You should also try to be honest about how much you feel you can critique the work in the **Review Confidence** level. Given the focus on neural networks for this semester, I would expect few *Low* confidence levels, but it's an option. For your **Format** section, I don't expect much to report since we will be using a set template, but be specific when reporting problems. For example, identify the page, paragraph, and line numbers where the problem resides, and give a specific example of how the problem should be fixed. Cover *all* formatting issues that you can find. The **Summary Statement** should provide your overall positive and negative critique of the paper and the justification for your **Rating**. You should always provide strengths and weaknesses, and you should back up any comment (positive or negative) with a clear example from the paper itself. If there is a negative criticism to be made, you should also provide a clear example of what would fix the issue(s) you have found. While these things are stated in *broad terms* in this section, you will use the **Major**, **Minor**, and **Discretionary** sections to enumerate all of the critical issues with the paper. **Major** issues are anything that allow a reviewer to doubt that the results of the project are conclusive or were adequately tested. **Minor** issues arise when clarification or additional demonstration is warranted, but the key results of the project would still remain valid. Finally, **Discretionary** revisions are more related to style, presentation, or *wish-list* requests related to the project.

Rubric

Formatting (7 points)

- Does your review match the requested format?
- Did you adequately review the formatting of the paper itself?

Summary Statement (5 points)

- Do your comments provide helpful insights that will strengthen the paper if addressed?
- Is it clear what the strengths/weaknesses of the paper are?

Revisions (18 points)

- Major – Did you address most/all of the major revisions the paper would require?
- Minor – Did you address most/all of the minor revisions the paper would require?
- Discretionary – Did you have some additional insights/questions to share with the authors?

Submission

This part of the project will be performed *independently* (not with your team). You should prepare your review as a PDF document. Keep in mind that *no team member names or identifying information* should be included in the review document. Submit your PDF review document to the assignment submission system under Project_Part_5: LINK.

Part 6 - Project Demo (60 Points)

Due: May. 3 @ 9:30am

Requirements

You should prepare a an example script set or notebook (or set of code/documentation materials) which provides a motivating example of your project while demonstrating your key code methodologies and your key findings/results. The demo should be **tutorial focused** in that a peer who has not had this course would be able to navigate to your team's Github repository page, and then follow a set of instructions from start to finish will minimal effort. This information should be included in a **Demo** section of the README.md file in your Github repository, but may link to other pages/sites/materials if needed (make a subdirectory named `demo` with the necessary components). The demo should mainly provide the nuts-and-bolts steps and instructions needed to run the demo (including steps to install the necessary Python packages), URLs for data sets, shell commands and information on how to use the demo materials, etc. However, it should also include a decent amount of guided documentation which explains the steps in the demo process in terms that help explain the motivation, methods, and results of your project to someone not familiar with neural networks.

Advice

I am looking for something similar to the tutorials I provide as part of the Homework and Open Lab assignments in this class. However, you will need to provide a little more documentation since I assume familiarity with the JupyterLab machines, and your demo users will not necessarily have access to these materials. You don't need to get bogged down in the minutiae of how to install jupyter notebook or juyter lab, but setting up a list of prerequisites with links to pages (from the jupyter team) which provide instructions on how to do this would be required.

Rubric

1. Do you utilize the README.md and linked materials to provide necessary prerequisite information (including installation help) at the start of the demo? (3 points)
2. Is the motivation for the demo clearly indicated to the user (why will it be helpful for the user to understand what your team has accomplished)? (15 points)
3. Does your demo provide a succinct set of steps which can be run within a 10-minute timeframe for unfamiliar users to run your demo and generate results? (22 points)
4. Are your results clearly presented back to the user in a way that allows them to understand what they mean? (13 points)
5. Do the results clearly back up the main aim(s) of your project? (7 points)

Submission

The user should be directed to how to start your demo from the README.md displayed at your team Github repository page. You should place all demo materials/files in an aptly named directory (`demo` or `Demo`). The user should need to step through a series of guided steps to complete the demo (i.e. you cannot just make a script called `demo.py` and have it just generate everything without user interaction). The demo will be utilized in class for the

presentation, so keep in mind that the steps will need to fit into a reasonable time window. You can provide option steps which load pre-trained weights or data sets to shorten the demo for the presentation when necessary. (Include this information in comments or documentation so that users outside the course will be able to do the same if desired.)

Part 7 - Project Presentation (35 Points)

Due: May. 3 @ 9:30am

Requirements

You should prepare a presentation using PowerPoint/PDF\Other Slides aimed at 18 minutes of content (20 minute slots). This will typically be no more than 9-10 slides for this amount of time. You should prepare a few slides (about 5 minutes) to introduce your project motivations and key aims, then the general strategy that your group took to address those project aims (include technical details when needed, but remember that you should spend more time talking about what you accomplished with your software than the software itself). You should then have each team member rotate and spend at least one minute each describing project contributions accomplished by she or he (about 6 minutes total). Finally, you should spend about 6-8 minutes showing off your demo materials, and how your demo explains the results from your project. Leave a minute or two for questions and the necessary transitions between group presentations.

Advice

Make sure to follow the requirements. The key aims of the presentation are to 1) show what each team member contributed to the project, and 2) show-off with an exciting demo. You should always interleave interesting results and lessons learned into this process, but *try to focus* on what **did** work instead of what **did not** work.

Rubric

1. Does your presentation clearly introduce the audience to your project and its motivations? (7 points)
2. Did you clearly and succinctly describe the project aims? (12 points)
3. Did each team member indicate his/her specific contributions to the project? (10 points)
4. Did your demo demonstrate the key elements of your project and address its key aims? (6 points)

Submission

An electronic, aptly named, version of your presentation (eg. Project Presentation.pdf) should be committed to the **project** repository for your team (eg. S21-team0-project) in the top directory of the repository.

Part 8 - Team Assessment (60 Points)

Due: May. 3 @ 11:00pm

Requirements

Complete and submit the Project Part 8 survey. This should be performed **independently** for each team member. You should not consult with one another in any way for this part of the team assignment since points are assigned individually in this category. In other words, working together on this part of the assignment constitutes Academic Misconduct under university and departmental policies and will result in at minimum penalty of a zero for this part of the assignment.

Advice

Resolving team issues often and early is important. Be proactive since your own assessment score as well as your teammates' scores will depend on setting clear expectations and delegation of tasks.

Rubric

1. Did you completely fill out the form in it's entirety? (5 points)
2. Do your percentage score totals for team expectations and your own expectations sum to 100%? (5 points)
3. Do your percentage score totals for team deliverable work and your own deliverable work sum to 100%? (5 points)
4. Do your scores reflect clear commitment to the work for the project? (25 points)
5. Do your scores reflect clear communication with the rest of your team? (20 points)

Submission

Complete your assessment document, and submit it as a single PDF to the course assignment system, Project_Part_8.

Part 9 - Team Allocation (40 Points)

Due: May. 3 @ 11:00pm

Requirements

Complete and submit the survey from Part 8 by the deadline above.

Advice

All teams have strengths and weaknesses, and since the assessment above will be graded these will inevitably come to light. Try to resolve issues early and often via communication. You may and should, if needed, request for the course instructor to assume the role of an anonymous faciliator to improve team communication.

Rubric

Points in this category are allocated to each team (not individually). In general, you still **must** submit your (complete) assessment above to receive any of the points for this part of the assignment, but these points will be the same for all team members who do so.

1. Did your team communicate clearly regarding work deliverables and expectations? (10 points)
2. Did the quality of your project suffer due to team communication issues? (10 points)
3. Did the quality of your project suffer due to team planning and/or time-related issues ? (10 points)
4. Did the quality of your project suffer due to technical issues? (10 points)

Submission

Contingent on Part 8 above.