CIS 522

Final Project

2/11/2020

1 CIS 522 Final Project Overview

Deep Learning has revolutionized the way the world operates today, from allowing us to perform real time object detection to assisting in medical diagnostics, surpassing human ability in some areas. As such, there's no better way to get a sense of what it's like to be a data scientist in the real world than to employ the skills you've learned in this class to a real-world, data-drive problem that excites you. You will work in a team of 3 to 4 students, and together will come up with a project, implementation, report, and succinct video describing your findings. The last time we offered this course, the students enjoyed the final project and found it rewarding – we hope that you will as well.

Learning Goals:

1. Carefully survey literature and applications to find an application that would benefit from deep learning or a knowledge gap that can be filled by using deep learning.
2. Create a short writeup to pitch your idea to others in the field (the TAs, in this case).
3. As a group, outline a detailed plan for how you will implement your idea. Plan the dataset you will use, the best modeling approach, the compute you will need, etc. Come up with milestones to achieve the project.
4. Obtain and explore the dataset.
5. Implement the model architecture that you think will do well on this problem and data.
6. Debug, refine, and improve your model.
7. Create insightful and engaging visualizations to summarize your findings.
8. Create a written report of your findings.
9. Create a video summary of your project.
10. Present your project to your classmates and other students.

2 Project Timeline

Deliverable	Assigned Date	Due Date
Project Abstract Proposal	2/11/20	2/18/20
Project Proposal / Team Formation	2/21/20	2/28/20
Milestone 1	2/28/20	3/20/20
Milestone 2	3/20/20	4/10/20
Milestone 3 + Code Submission	4/10/20	4/24/20
Poster Presentation + Paper Presentation	4/24/20	4/28/20

Above is the timeline for the final project. Each of the components of the final project is described in more detail in the sections below.

3 Project Abstract Proposal

Description: Your first deliverable will be a short (300 word maximum) abstract on the project idea you would like to work on for the course. To come up with a project idea, we would like you to survey existing applications and literature in a domain of interest to you (i.e. Natural Language Processing, Reinforcement Learning, Computer Vision, Theoretical Deep Learning), find a potential application or knowledge gap you'd like to work on, and create a 300 word (max) pitch in the form of an abstract to convince others as to why this would be a cool project (in this case, you will be convincing the TAs). **This is an individual deliverable i.e. every student must submit an abstract.**

Requirements: Your abstract must contain mention of each of the following components

1. Motivation for the broader question (i.e. why care about this topic)?
2. What is already known or done in this domain?
3. What needs to be done differently? (i.e. knowledge gap or problem statement)
4. What modeling approach will your project use to address this gap/problem?
5. What would be the expected outcome of your project (i.e. tying it all back to the original knowledge gap, problem statement, and motivation)
6. Cite 3 related papers or models that are key for background reading

Proposal Rankings: The TA staff will go through all of the project abstract proposals and rank them on a variety of metrics (including how feasible the project is, how interesting the project is, etc.) and curate a list of **35** projects and release this list on Piazza. By the next milestone, you will look at these projects, and form a team amongst yourselves (with the person who submitted the proposal being on the team by default). **Note:** Whether or not your project is selected **will not** affect your proposal grade provided your abstract proposal meets all the requirements above.

4 Project Proposal + Team Formation

First, you must join/create a team of 3-4 other CIS 522 students, from one of the 40 curated projects that the TA team has selected. Once you've created your team, your team will deliver a more in-depth version of the previously submitted abstract. It's fine if things have changed since the original abstract, especially with regards to the method! The spirit of the idea must be similar, however.

Requirements: This proposal must contain the following elements

1. All the team members' names (3-4 students per group)
2. A description of the dataset being used (where it's from or how it will be made) as well as a sample of the data (i.e. 5 rows of the dataset if it's in a dataframe-like format or an appropriate visualization of 5 samples from the dataset)
3. A description of your hypothesis (i.e. what do you expect to happen?)
4. A description of your inductive biases (i.e. what is your rationale / intuition for why you think your hypothesis is true)
5. A related works section detailing papers relevant to your project as well as their findings / relevance to your project idea.
6. An in depth description of your expected implementation / model (**note:** it is completely fine if this does not end up being your final model, we expect that things will change throughout the course of the project)
7. A timeline up until the project report deadline (final deadline is Tuesday, April 28th 2020) with three distinct milestone deliverables that a TA can follow up on. Your milestones should be specific to your project and realistic. The milestone meetings with a TA must be scheduled before the following dates:
   (a) First milestone meeting by March 20th
   (b) Second milestone meeting by April 10th
   (c) Third milestone meeting by April 24th (you will submit your code at this date!)

5 Final Project Submission + Requirements

The Final Project will compose of three parts:

1. Final Project Report (300 pts)
2. Final Project Video (150 pts)
3. Novelty (50 pts)
4. Code Submission (50 pts)

Aside: As an aside, we want to clarify that we know projects will change pretty significantly from the initial final project proposal that you submitted. **This is fine and also expected!** We would be pretty surprised if everything went exactly as planned in your project proposal. Any time you have major changes, feel free to reach out to your TA to get their advice and discuss amongst your team what the right pivot for your project should be. You won't be penalized for this – as we'll explain shortly, we'll be grading more-so on the process of experimentation and not the actual results of the experiment.

5.1 Final Project Report (300 pts)

Overview: For those that have written academic papers previously, we expect the Final Project Report to be structured similarly to an academic paper with the following key distinctions:

1. We expect the bulk of your paper to be about the process (i.e. what different things did you try? How effective were they?) and **not** about the actual results. We know that many groups are trying an approach that is novel, and therefore may not work. **That's completely fine**. You'll still get full credit (as you'll see from the rubric below) as long as you document your process.
2. There will be no need to have an Abstract for your final project.

Typesetting / Formatting: The only requirement that we have is that the Final Project Report be written in LaTeX. We highly recommend that you use a LaTeX from your project's domain (i.e. for CV projects you would use the CVPR LaTeX template, for NLP you would use the ACL LaTeX template).

Sections of the Report: The Final Project Report should be composed of the following sections:

5.1.1 Introduction (30 pts)

In the introduction you should give a high level overview for the topic that your paper will be exploring. As long as this section is written clearly and serves as an introduction to the rest of the report, you should get full points for this.

5.1.2 Related Work (50 pts)

Here you will discuss academic papers that are relevant to your group's final project. You'll briefly discuss their methods / approaches to give enough context to motivate your own approach. As long as this section

5.1.3 Methods (75 pts)

In this section you will detail your deep learning methods that were tried over the course of the final project. To receive full points, your group should have implemented and evaluated the following:

1. A non deep learning baseline (can for instance be a classical machine learning approach like using Logistic Regression / SVM for classification).
2. A deep learning baseline (like a standard feed forward neural network, CNN, LSTM, etc.)
3. An advanced deep learning approach (i.e. your novel approach for your final project).

You should be detailed about each of these approaches, listing out specifics of your chosen model architecture (i.e. how many layers? What learning rate did you choose? What does your architecture look like? How did you do hyperparameter tuning?). Imagine that someone were to read your paper and try to exactly reproduce your results without looking at your code at all – would they be able to do so with the description present in your Methods section?

5.1.4 Analysis (75 pts)

Analyze the methods that you laid out in the previous section. Compare the efficacies of your non deep learning baseline vs the deep learning baseline vs your advanced deep learning approach. Which model works best? Why? If you implemented additional approaches, but found that they did not succeed, be sure to include their analyses here as well. Your analysis must contain at least the following:

1. If you have relevant performance metrics, the performance metrics for each of your different approaches and an analysis of how they compare.
2. Visualizations. Provide in your report your training loss curve / other relevant visualizations that help aid in your analysis.
3. A discussion about why one model performs better than another. Use your visualizations and any other evidence from your experimental results to explain why this is the case.

Note that like we said earlier, your advanced deep learning approach / novelty for your project does **not** have to outperform your deep learning baseline / non-deep learning baseline. As long as you document the different approaches, that's sufficient.

5.1.5 Discussion (50 pts)

Based on your results and work done so far, what further work do you think should be done with regards to your project? What other approaches would you try if you had more time? What conclusions do you think you can draw from your results?

5.1.6 References (20 pts)

This section is a bibliography for your paper. If you use a template from a major conference like NeurIPS, CVPR or ACL, it should be very easy to produce the references for your final report. If you choose not to use such a template, feel free to cite the papers in whatever format you would like.

Note: As you can see – most of these sections are graded based on completion. As long as you follow the guidelines, you should be able to score well on this project. The only portion of your grade relevant to the technical difficulty of your project / the novelty of it is in the novelty section below.

5.2 Final Project video (150 pts)

Your group should produce a succinct 5 minute video about your project that shows off the cool work your group has done for the semester! Since this is the first time this class is being run, we don't have any example videos to show you, but a great example of what we're trying to shoot for are oral presentations at conferences. For instance, click **here** for the oral presentation for a Best Paper finalist of CVPR 2019. Your video should consist of the following components:

1. A brief introduction to your final project.
2. Related Work
3. The approaches tried
4. An analysis of the approaches tried and how they compare
5. A discussion of the conclusions that can be drawn from your experiments.

As you can see, the final project video is more or less a condensed version of your final project report into 5 minutes! We'll be sharing your videos with the rest of the class so feel free to be creative! We'll be grading the final project video with the following breakdown:

1. **Clarity (50 pts)**: Does the video clearly describe the purpose of the project, the approaches tried, and the results / conclusions from those approaches.
2. **Content (50 pts)**: Does the video contain sufficient technical description such that one could understand the approach from a technical perspective (should **not** contain minutiae such as hyperparameters, but should discuss the general network architecture, etc).
3. **Analysis (50 pts)**: Is the analysis well presented through the video format and thorough? Do they present a thorough analysis of how the different approaches compare in a way that is easily understandable through the video format? What conclusions do they draw and how well do they explain the rationale for these conclusions in the video?

5.3 Novelty / Technical Difficulty (50 pts)

The novelty for your project is graded on a scale from 1-5 with the following criteria:

1. **10 / 50 pts** The project has no novelty and is a straightforward application of a simple deep learning approach already done in class. For instance, just implementing / tuning a CNN to perform a classification on an image would fit this criteria.

2. **20 / 50 pts** The project has no novelty but is an implementation of a more advanced deep learning technique already done in class. For instance, implementing and tuning a DCGAN on a dataset would fit this criteria.

3. **30 / 50 pts** The project has no novelty but is an implementation of a more advanced deep learning technique not already done in class. For instance, implementing and tuning a WGAN on a dataset, or implementing a perceptual loss would fit this criteria.

4. **40 / 50 pts** The project has some aspect of novelty and is relatively technically difficult, however, the novel approach did not lead to better results. For instance, if you implemented an attention model but had a novel approach of weighted the previous hidden state vectors, but the novel approach did not end up working

5. **50 / 50 pts** The project has aspects of novelty, is relatively technically difficult, and the novelty worked! Honestly just go publish a paper at this point (just kidding, it doesn't have to be enough novelty that would actually constitute a paper, but still should have some aspects of novelty).

5.4 Code Submission (50 pts)

Code submission can either be a jupyter/colab notebook (preferred) or a python project.

5.4.1 Jupyter notebook requirements

Distinct sections of the notebook to make the code readable. For instance, a good structuring could be (although by no means do you have to actually stick to this structure, this is just one possible structure):

Section 0: Imports / Installs / Mounting Google Drive / Tensorboard Setup In this section you would pip install all necessary packages as well as import necessary modules for your project. You would also mount to Google Drive and set the file paths to be used by the rest of the project. Also in here could be your tensor board setup as well as any helper methods used by the rest of your project.

Section 1: Data Loading / Cleaning / Processing / Creating train test splits Here you can load your CSV (/ whatever the data format is) and perform any preprocessing to the data. For instance, if you're doing NLP, here you can lemmatize / tokenize your text, set up the TorchText Fields, TabularDataset, etc etc.

Section 2: Dataset / DataLoader Setup Essentially all projects will require some custom Dataset to be defined for the data that you want loaded in and to initialize a DataLoader for that dataset. The custom dataset / data loader depending on the application can be relatively nontrivial / take quite a bit of code to set up, so having a section for this setup could be a good idea for your project.

Section 3: Network Model / Network Initialization / Loss Here you define your network model(s) and initialize them along with your optimizer(s). Also if you have a custom loss function that you must define, define it in this section.

Section 4: Training / Validation / Testing Loop Here you define your training / validation / testing loop for your network model. For most applications, a good way of using the validation set is to, per epoch, calculate the average loss / accuracy for your validation set and report that. Note that accuracy may mean very different things depending on your project! Be sure to figure out what performance metric makes the most sense for your project / domain.

Section 5: Analysis of Outputs Here you can analyze your trained model by plotting confusion matrices to find out why certain errors occurred, plotting histograms, cluster plots, etc etc. The goal would be to use this section to really analyze your model's performance, and figure out why certain errors were occurring (for instance, does it indicate a class imbalance issue? is there a pattern to the errors? is one class just really hard to predict?)

5.4.2 Python Project Requirements

We don't recommend this if possible, but we know there may be some groups for which their project doesn't lend themselves well to a Jupyter notebook. In this case it's also perfectly fine to submit a Python project, however, you must make sure to have comments in your code and practice good style practices.

5.4.3 Requirements for All Submissions

1. Some form of documentation for how to run the project. For instance, how do we download/where do we put the data? Is there any custom installation we have to do? We should be able to train your model using the instructions you provide. Being able to run the project using your documentation will be worth 25 / 50 pts for this section.

2. Proper python style practices. For an example of such a guideline, feel free to look at Google's Open Source Python guidelines: http://google.github.io/styleguide/pyguide.html We won't be very strict on style, but if your style is bad to the point of being difficult to read, then we will deduct points for this. Style is subtractive, so particularly bad style will be given a -10pts for the project.

3. Having a layout of your code that is easy to navigate and understand will be worth 25 / 50 points. Again, feel free to modify the structure above to whatever works best for your project (or use your own structure) – you should just make sure that your code is easy to navigate / read / understand.