Final Project Guidelines

STATS 315, Fall 2022
Yixin Wang
University of Michigan

The final project is an integral part of this course. There is no better way to learn about deep learning than by applying it to a research question of your own. The goal of the final project is for you to implement an entire deep learning pipeline yourself: data loading, model design, training, evaluation, and analysis. The final project is an individual project.

You have a lot of freedom in choosing a topic for your final project. The only criterion is that it substantially involves applying deep learning models to a real-world problem. You choose a dataset, select an interesting question about it, and address the question with deep learning.

The project involves three assignments: project proposal, project milestone, and project report. For each, please use a 12-point font, and 1-inch margins. Page limits are without figures; include as many pages of figures as needed. We offer a LaTex template; you are encouraged to use it but not required.

Project Proposal

The project proposal is an abstract (i.e. a paragraph) that imagines the completed project. We understand that your project will evolve and change over the semester, but writing an abstract early is a good way to plan and think about what it would mean to successfully complete it. (Kaggle is a good place to find interesting datasets that may be suitable for deep learning.)

We encourage you to refer to computer science conferences (such as *Neural Information Processing Systems* and *International Conference of Machine Learning*) or journals (such as *The Annals of Applied Statistics*, *Journal of the American Statistical Association*, *Journal of Machine Learning Research*) to get a sense of how to write an abstract.

Project Milestone

The project milestone describes the problem you are addressing and discusses some preliminary results. Include what you have completed and what you plan to finish by the end of the semester. As a guideline, the project milestone should (at least) answer the following questions.

- Clarify the purpose of the prediction task:

    - What is the problem of interest?
    - What is the final goal you want to achieve?

- To the best of your knowledge, explain the data as thoroughly as possible:
    - How is the data collected?
    - Is the data a simple random sample from some population?
    - Are there any defects in the data? If so, how might they affect your analysis?
    - If you were able to collect the data, what would you have done differently?
    - Are you going to account for the data collecting process in your analysis?
    - Briefly describe the data, including its size, structure, and source.
    - If you are not going to use real data, explain how you are going to generate "realistic" synthetic data that represents your knowledge of the problem.

- Draft a plan for data analysis:
    - What deep learning model are you going to use for the prediction?
    - Have you done any preliminary analysis of the data?
    - What is your timeline for (1) proposing and fitting a preliminary deep learning model, (2) examining initial results and adjusting models further if needed, (3) summarizing and writing a final report?
    - What are the (potential) difficulties with your data analysis?

The project milestone is 1–3 pages long.

Project Report

The project report is due at the end of the semester a. The report should have a similar format to that of an academic paper.

The project report should (at least) include

- **Introduction:** Clear description of the problem. Describe the problem you are trying to solve, why it is important or useful, and summarize any important pieces of prior work that you are building upon.
- **Dataset:** Clear description/visualization of the data. Describe the dataset or datasets you are working with. Show examples from your dataset to give a sense for the types of data it contains. If you collected or constructed your own dataset, explain the process you used to collect the images and labels, and why you made the choices you did in the data collection process.

- **Method:** Clear and thorough description of statistical analysis. Describe the method you are using; this may also contain parts of the implementation of your model, loss function, or other components along with sanity checks to ensure that those components are correctly implemented.
    - You are encouraged to compare multiple architectures. Training the same architecture in two different ways (e.g., with a different learning rate or optimizer) counts as one model. You may need to experiment with different optimizers, learning rates, L2 regularization strengths, and data augmentation strategies. You should train at least 6 different models, and report their accuracy on both the train and validation sets.
    - You should show training curves for your models. You should plot the training loss per iteration, as well as the accuracy of the model on the train and validation sets every epoch (pass through the training data). You may try using additional techniques like test-time augmentation or model ensembling to improve upon the prediction performance of your individual models.
    - You should also include a table that summarizes the final train and validation accuracies of all your models. You can be creative in exactly what models you choose to train.
- **Experiments:** Clear and thorough explanation of experiments and interpretation of results. Describe the experiments you did, and key results and figures that you obtained. This may interleave explanations of the experiments you run and figures you generate as a result of those experiments.
    - You should analyze the models you trained to try and gain insight into how well they are working and where they make mistakes. You should analyze at least two different models: your best-performing model and at least one other of your choice.
    - For example, you may show qualitative examples of data points that were both correctly classified/predicted and incorrectly classified/predicted. You can show both data points hand-picked by you to showcase interesting features of your model, and randomly-chosen data points to give a better sense of the average performance of your model.
    - You may also show confusion matrices of your predictions if you do classification, a 2D matrix of predicted category vs ground-truth category, where each entry shows the fraction of validation-set data points falling into this situation; this result demonstrates the kinds of mistakes that your model makes.
    - You may also look up and use one or more neural network interpretation tools to give examples of what portions of data points are used by the model for prediction decisions.
    - After all your analysis, you should run your best-performing model on the test set. If your model performs very differently on the validation and test sets, then

you should attempt to explain why that might be the case.

- **Conclusion / Future work:** What did you learn in doing this project? What are the shortcomings or failure cases of your work? If you had more time or resources, how would you continue or expand upon the work you have already done?

You must submit your final report and its appendix in a single PDF file. Notebooks or markdowns with annotated code which generate the results (tables, figures, etc.) from the final project should be included as an appendix of the final report. You must run all cells in your notebook to receive credit; we will not rerun your notebook. You can include extra appendices (of any length) to which you can banish any details. (We may not read the appendix when grading your work, except to check the code.)

The project report is 3–5 pages long, excluding figures or references.

The deadline of the final project report is strict, and late days cannot be applied. All late final projects receive a score of zero. In case of requesting exceptions due to severe medical reasons, a doctor's note and a signature from your graduate (or undergraduate) advisor is needed.

Policy about Open Source Code

While there are many open source implementations for the problems described below, you should not use them for your implementation. You may refer to existing open-source implementations if you are confused about a specific detail, but you should neither import from nor copy code directly from existing implementations. Your implementation should only use standard libraries for scientific computing in python (e.g. numpy, scipy, matplotlib, pytorch, torchvision, tensorflow, JAX, etc). If you have questions about whether you should or should not use an existing library, please ask about it on Piazza.

Project Evaluation

We evaluate the project on ambition, significance, originality, technical depth, results, relevance, and writing quality. Two good books about writing are Strunk Jr and White (2007) and Williams (1990); see the reference section for more information on these books.

Any projects containing plagiarized code or text will receive an automatic zero. Projects that are irrelevant to course content (i.e., not substantially focused on applying deep learning to a data set) will lose 40% automatically.

Aside from these penalties, the grading rubric is as follows:

Category	Individual Requirements	Pts	TOTAL Pts
Problem understanding & description	● Effectively motivates and explains problem	2	5
	● Describes data and its source	2	
	● Identifies valuable goals for data analysis/modeling	1	
Quality of data analysis **NOTE:** The standards for this section are higher for easier projects. If your project uses a frequently analyzed tabular data set, for instance, then to earn full marks you'll need to fit very sophisticated models and perform a detailed analysis that reveals interesting and unique insights about the dataset.	*Deep learning models*	*(25)*	50
	● Successfully trained 6 interesting and diverse models	10	
	● At least one model considered is built without the keras Sequential() class (or a similar function/class that abstracts away the relationship between layers) and was implemented appropriately	10	
	● Model architectures are explained clearly enough that another student could replicate them	3	
	● Model training procedure is reasonable	2	
	Model comparisons & analyses	*(25)*	
	● Selected appropriate testing/validation metric(s)	3	
	● Calculated testing/validation metric(s) for all 6 models and compared them in a table	6	
	● Includes training curve figures	6	

	• Performs interesting and detailed analysis of two models (e.g., failure cases, confusion matrices) and/or model interpretation)	10	
Presentation of results	• Figures are appropriately labeled with axis labels, legends, and titles	3	10
	• Figures and tables are numbered and have clear captions that explain the contents and the conclusion	3	
	• Visualizations are used judiciously throughout the report to effectively communicate findings	4	
Interpretation of results	• Conclusions are well supported by the results	10	15
	• Discusses results in terms of the initial research question	5	
Code	• Code was submitted as an appendix	3	10
	• Code was executed before submission and contains all results displayed in the report	4	
	• Code is logically organized and easy to read	3	
Writing	• Report is well organized	2	10
	• Report has few or no substantial grammatical errors	3	
	• Writing is clear and concise	3	
	• Report length is about 3–5 pages excluding figures and captions; reports that are too long will be penalized	2	

References

Strunk Jr, W. and White, E. B. (2007). *The Elements of Style Illustrated*.

Penguin. Williams, J. (1990). Toward clarity and grace. *Chicago: The University of Chicago*.