Intro ML Project

Fraida Fund

For the project component of this course, you will study a piece of recently published work in machine learning by: learning about your the work described in the paper, replicating some existing version of it (using a Python notebook or source code that is provided for you), and then extending that provided notebook or source code to highlight some element of the work.

Goals

I assign this project format because as you move on from this introductory course, it's important to:

- Engage with high-quality research in the field of machine learning.
- Practice building on your foundation in machine learning fundamentals, to learn new concepts and techniques.

What to expect

To learn about the work that is the topic of your project, you should expect to read the primary research paper or other academic material describing the work. You may need to read additional papers or other materials to fill in gaps between the introductory material in the lectures, and the project topic. For example, if your topic involves a kind of neural network called a *transformer*, you will probably have to learn about encoders and decoders, attention, and self-attention before you can begin to understand the transformer. You can ask for learning resources on Ed if you're having trouble finding high-quality learning material on any particular topic.

To replicate the work, you'll use a Python notebook or open-source Python code that will be given to you. Depending on the specific topic, you may find that the notebook provided to you runs on Colab without any changes necessary. Other students may have to do some more work to: retrieve data into the Colab workspace, install Python libraries or specific versions of Python libraries, make small changes to the code to work with different library versions, or similar modifications.

Finally, you'll extend the provided notebook or source code to highlight your understanding of the work. You'll have to decide the best way to extend the work, but it's important that you do something that shows your understanding. Some examples:

- You might show the output of intermediate steps of a multi-step process, to show how the process works.
- If the paper that describes your method discusses some of its strengths and weaknesses, you might use specially prepared test samples to show how the model does well on test samples that fall within its "strengths" and poorly on test samples that fall within its "weaknesses".
- You might use a completely different type of test data, explain why you would expect the model to perform well (or not!) on that type of data, and show what you could do to improve the model performance on that type of data.
- You might compare and contrast different methods for a similar problem using the same data, and explain any differences in their performance using your understanding of the methods.
- If your model uses an interesting and non-obvious loss function, or is evaluated by an interesting metric (e.g. ROUGE, BLEU), you can break down the steps involved in computing it, and show what

it computes for selected "good" and "bad" samples to explain why it works.

Important note: hyperparameter tuning does *not* show anything about your understanding of this particular work. You should *not* bother with hyperparameter tuning, and you should *not* consider it an "extension" of the work.

Note that many of these "extensions" will work even with a pre-trained model! In many cases, you will not need to actually train a model from scratch.

The process

Selecting a project

You'll choose your project from the following list (open with your NYU Google account) on a first-come, first-served basis. (Two students cannot do the same project.)

https://docs.google.com/document/d/1UERHzBDwU33_kuiLwVFIL9jLuuKtg4B9UiZSLAz1Df0/

To "claim" your project:

- In the Google Doc, find the list item for your selected project, and make sure it is not already "claimed".
- Highlight the entire list item, and add a comment with the names and email addresses of your entire "team". A "team" can be one student or two students.
- Do not mark the comment as "resolved".

I have a project topic, now what?

0. Quick review of the paper A quick scan of the paper will help you get the main points of what it's all about, and also help you identify relevant items to look out for when running the accompanying code.

1. Start by getting some baseline code running (on Colab) All of the projects include a link to some code - one or more notebooks, Github repositories, blog posts with code snippets included, or other source code that you can use as a baseline for your project.

A good first step is to get some baseline code running on Colab. Some of you have a notebook that will run as-is without any changes, so you don't have to do anything to get your baseline code running. Some of you will need to make a few small changes, like changing the Tensorflow version from 2.x to 1.x, or installing some packages, in order to get your baseline code to run. Some of you will need to make more substantial changes.

Why should this be your first step? If you can't get your baseline code working, you should reach out to me for help as soon as possible, so that I can work with you to get something running. If we can't get your baseline code working together, then I'll help you get started on a different, related project instead.

If you spend a lot of time and effort on your project topic before you get some code running, that time and effort might be wasted if you end up having to switch topics.

2. Learn about your project topic Once you have your baseline code running, a good next step is to do a deep dive into your project topic and learn much more about it. Read the paper where the topic is introduced, and any other materials (blog posts, etc.) that I recommended. You should also seek out additional high-quality learning resources on your own (make sure to keep track of all your sources, so that you can cite them properly in your report!)

Look for details like:

- How exactly does this technique work?

- What other techniques are there for similar problems? What is special about this technique, compared to the others?
- What kind of problems/data would this technique be best suited for? What kind of problems/data would it not perform well on? Why?
- Are there other techniques that perform better than this one under some circumstances? If so, is there any circumstance in which this technique is still the best?

You can use your baseline code to help you understand the topic, too. Look at the source code, and try to understand how it connects to the details you've read about how the technique works. You may add extra visualizations or other output to your baseline code to help you understand the technique.

3. Plan to extend your work After you have some baseline code running and you have a good grasp of the topic/technique, you are in a good position to extend the baseline in a way that shows your understanding of the topic.

A good original contribution is one that shows that you understand the technique, how it is used, and what its strengths and weaknesses are. For example, you might apply it to a problem it is especially well suited for (and explain why it is!), compare it to another technique that is used for similar tasks (and make sure to explain the results!), show how it works with some carefully selected test samples that are designed to highlight its strengths and weaknesses, etc.

If you're not sure about your planned extension, feel free to ask for feedback!

4. Prepare your presentation and notebook submission You'll have to submit a pre-recorded presentation about your project, and a notebook with text (in your own words), code, and images. More details will be shared about these items in a separate document.

Attribution will be an important grading criteria for both of these. Make sure that as you work, you keep careful track of what is your own (ideas, text, code, images) and what you use or adapt from other places (including the materials given to you). For items that are not original, but that are used or adapted from other places, make sure to keep track of the original source and any changes you made.

Getting help

This project asks you to go a few steps farther than what we have covered in class. It's supposed to be challenging, but if you get stuck, you don't have to struggle alone!

You can post on Ed to get help with conceptual questions (Example: "I can't figure out what attention layers do. All of the online resources I can find - I listed them below - are about attention layers for NLP problems, but my problem is about understanding images, and I don't know what attention layers do in that context.")

You can also post on Ed to get help with technical questions (Example: "I know I need to install some specific package versions to get my notebook to run, but I can't figure out which! These are all the things I have tried...")

And, you can post on Ed to get help with planning how to extend the existing work (Example: "My baseline code does pose estimation on a YouTube video. Here's a link to the notebook. Would running this on a different YouTube video be a good extension?")

When you post on Ed, feel free to include a link to your draft Colab notebook, if relevant, but make sure to adjust the sharing permissions so that I can see it.