This assignment is due on **Tuesday, May 25 2021** at 11:59pm PST.

Starter code containing Colab notebooks can be downloaded here.

Setup

Please familiarize yourself with the recommended workflow before starting the assignment. You should also watch the Colab walkthrough tutorial below.

Note. Ensure you are periodically saving your notebook (`File -> Save`) so that you don't lose your progress if you step away from the assignment and the Colab VM disconnects.

While we don't officially support local development, we've added a **requirements.txt** file that you can use to setup a virtual env.

Once you have completed all Colab notebooks **except** **`collect_submission.ipynb`**, proceed to the submission instructions.

Goals

In this assignment, you will implement language networks and apply them to image captioning on the COCO dataset. Then you will explore methods for visualizing the features of a pretrained model on ImageNet and train a Generative Adversarial Network to generate

images that look like a training dataset. Finally, you will be introduced to self-supervised learning to automatically learn the visual representations of an unlabeled dataset.

The goals of this assignment are as follows:

- Understand and implement RNN and Transformer networks. Combine them with CNN networks for image captioning.
- Explore various applications of image gradients, including saliency maps, fooling images, class visualizations.
- Understand how to train and implement a Generative Adversarial Network (GAN) to produce images that resemble samples from a dataset.
- Understand how to leverage self-supervised learning techniques to help with image classification tasks.

You will use PyTorch for the majority of this homework.

Q1: Image Captioning with Vanilla RNNs (30 points)

The notebook `RNN_Captioning.ipynb` will walk you through the implementation of vanilla recurrent neural networks and apply them to image captioning on COCO.

Q2: Image Captioning with Transformers (20 points)

The notebook `Transformer_Captioning.ipynb` will walk you through the implementation of a Transformer model and apply it to image captioning on COCO.

Q3: Network Visualization: Saliency Maps, Class Visualization, and Fooling Images (15 points)

The notebook `Network_Visualization.ipynb` will introduce the pretrained SqueezeNet model, compute gradients with respect to images, and use them to produce saliency maps and fooling images.

Q4: Generative Adversarial Networks (15 points)

In the notebook `Generative_Adversarial_Networks.ipynb` you will learn how to generate images that match a training dataset and use these models to improve classifier performance when training on a large amount of unlabeled data and a small amount of labeled data. **When first opening the notebook, go to `Runtime > Change runtime type` and set `Hardware accelerator` to `GPU`.**

Q5: Self-Supervised Learning for Image Classification (20 points)

In the notebook `Self_Supervised_Learning.ipynb`, you will learn how to leverage self-supervised pretraining to obtain better performance on image classification tasks. **When first opening the notebook, go to `Runtime > Change runtime type` and set `Hardware accelerator` to `GPU`.**

Extra Credit: Image Captioning with LSTMs (5 points)

The notebook `LSTM_Captioning.ipynb` will walk you through the implementation of Long-Short Term Memory (LSTM) RNNs and apply them to image captioning on COCO.

Submitting your work

Important. Please make sure that the submitted notebooks have been run and the cell outputs are visible.

Once you have completed all notebooks and filled out the necessary code, you need to follow the below instructions to submit your work:

1. Open `collect_submission.ipynb` in Colab and execute the notebook cells.

This notebook/script will:

- Generate a zip file of your code (`.py` and `.ipynb`) called `a3_code_submission.zip`.
- Convert all notebooks into a single PDF file called `a3_inline_submission.pdf`.

If your submission for this step was successful, you should see the following display message:

```
### Done! Please submit a3_code_submission.zip and a3_inline_submission.pdf to Gradescope. ###
```

2. Submit the PDF and the zip file to Gradescope.

Remember to download `a3_code_submission.zip` and `a3_inline_submission.pdf` locally before submitting to Gradescope.