This assignment is due on **Monday, May 02 2022** 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 practice writing backpropagation code, and training Neural Networks and Convolutional Neural Networks. The goals of this assignment are as follows:

- Understand **Neural Networks** and how they are arranged in layered architectures.

- Understand and be able to implement (vectorized) **backpropagation**.
- Implement various **update rules** used to optimize Neural Networks.
- Implement **Batch Normalization** and **Layer Normalization** for training deep networks.
- Implement **Dropout** to regularize networks.
- Understand the architecture of **Convolutional Neural Networks** and get practice with training them.
- Gain experience with a major deep learning framework, such as **TensorFlow** or **PyTorch**.
- Explore various applications of image gradients, including saliency maps, fooling images, class visualizations.

Q1: Multi-Layer Fully Connected Neural Networks

The notebook `FullyConnectedNets.ipynb` will have you implement fully connected networks of arbitrary depth. To optimize these models you will implement several popular update rules.

Q2: Batch Normalization

In notebook `BatchNormalization.ipynb` you will implement batch normalization, and use it to train deep fully connected networks.

Q3: Dropout

The notebook `Dropout.ipynb` will help you implement dropout and explore its effects on model generalization.

Q4: Convolutional Neural Networks

In the notebook `ConvolutionalNetworks.ipynb` you will implement several new layers that are commonly used in convolutional networks.

Q5: PyTorch on CIFAR-10

For this part, you will be working with PyTorch, a popular and powerful deep learning framework.

Open up `PyTorch.ipynb`. There, you will learn how the framework works, culminating in training a convolutional network of your own design on CIFAR-10 to get the best performance you can.

Q6: Network Visualization: Saliency Maps, Class Visualization, and Fooling Images

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.

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 `a2_code_submission.zip`.
- Convert all notebooks into a single PDF file.

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

```
### Done! Please submit a2_code_submission.zip and a2_inline_submission.pdf to Gradescope. ###
```

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

Remember to download `a2_code_submission.zip` and `a2_inline_submission.pdf` locally before submitting to Gradescope.