

CS231n Convolutional Neural Networks for Visual Recognition

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 these models on data
- gain experience with a major deep learning framework, such as **TensorFlow** or **PyTorch**.

Setup

Get the code as a zip file [here](#).

You can follow the setup instructions [here](#).

If you performed the google cloud setup already for assignment1, you can skip this step and use the virtual machine you created previously. (However, if you're using your virtual machine from assignment1, you might need to perform additional installation steps for the 5th notebook depending on whether you're using Pytorch or Tensorflow. See below for details.)

Some Notes

NOTE 1: This year, the `assignment2` code has been tested to be compatible with python version `3.7` (it may work with other versions of `3.x`, but we won't be officially supporting them). You will need to make sure that during your virtual environment setup that the correct version of `python` is used. You can confirm your python version by (1) activating your virtualenv and (2) running `which python`.

NOTE 2: As noted in the setup instructions, we recommend you to develop on Google Cloud, and we have limited support for local machine configurations. In particular, for students who wish to develop with Windows machines, we recommend installing a Linux subsystem (preferably Ubuntu) via the [Windows App Store](#) to streamline the AFS submission process.

NOTE 3: The submission process this year has **2 steps**, requiring you to 1. run a submission script and 2. download/upload an auto-generated pdf (details below.) We suggest ***making a test submission early on*** to make sure you are able to successfully submit your assignment on time (a maximum of 10 successful submissions can be made.)

Q1: Fully-connected Neural Network (20 points)

The IPython notebook `FullyConnectedNets.ipynb` will introduce you to our modular layer design, and then use those layers to implement fully-connected networks of arbitrary depth. To optimize these models you will implement several popular update rules.

Q2: Batch Normalization (30 points)

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

Q3: Dropout (10 points)

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

Q4: Convolutional Networks (30 points)

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

Q5: PyTorch / TensorFlow on CIFAR-10 (10 points)

For this last part, you will be working in either TensorFlow or PyTorch, two popular and powerful deep learning frameworks. **You only need to complete ONE of these two notebooks.** You do NOT need to do both, and we will *not* be awarding extra credit to those who do.

Open up either `PyTorch.ipynb` or `TensorFlow.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.

NOTE 1: The PyTorch notebook requires PyTorch version 1.0, which comes pre-installed on the Google cloud instances.

NOTE 2: The TensorFlow notebook requires Tensorflow version 2.0. If you want to work on the Tensorflow notebook with your VM from assignment1, please follow the instructions on [Piazza](#) to install TensorFlow. New virtual machines that are set up following the [instructions](#) will come with the correct version of Tensorflow.

Submitting your work

There are **two** steps to submitting your assignment:

1. Run the provided `collectSubmission.sh` script in the `assignment2` directory.

You will be prompted for your SunetID (e.g. `jdoe`) and will need to provide your Stanford password. This script will generate a zip file of your code, submit your source code to Stanford AFS, and generate a pdf `a2.pdf` in a `cs231n-2019-assignment2/` folder in your AFS home directory.

If your submission for this step was successful, you should see a display message

```
### Code submitted at [TIME], [N] submission attempts remaining. ###
```

2. Download the generated `a2.pdf` from AFS, then submit the pdf to [Gradescope](#).

 [cs231n](#)

 [cs231n](#)

karpathy@cs.stanford.edu