

CS231n Convolutional Neural Networks for Visual Recognition

Course Website

Assignment 2

This assignment is due on **Friday, April 30 2021** at 11:59pm PST.

Starter code containing Colab notebooks can be [downloaded here](#).

- [Setup](#)
- [Goals](#)
- [Q1: Multi-Layer Fully Connected Neural Networks \(16%\)](#)
- [Q2: Batch Normalization \(34%\)](#)
- [Q3: Dropout \(10%\)](#)
- [Q4: Convolutional Neural Networks \(30%\)](#)
- [Q5: PyTorch/TensorFlow on CIFAR-10 \(10%\)](#)
- [Submitting your work](#)

Setup

Please familiarize yourself with the [recommended workflow](#) before starting the assignment. You should also watch the Colab walkthrough tutorial below.

CS231n Google Colab Assignment Workflow Tutorial



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**.

Q1: Multi-Layer Fully Connected Neural Networks (16%)

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 (34%)

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

Q3: Dropout (10%)

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

Q4: Convolutional Neural Networks (30%)

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

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

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.** While you are welcome to explore both for your own learning, there will be no extra credit.

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.

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.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.zip and the pdfs to Gradescope. ###
```

2. Submit the PDF and the zip file to [Gradescope](#).

Remember to download `a2.zip` and `assignment.pdf` locally before submitting to Gradescope.