Introduction to Computer Vision (83551) - Assignment 1

The goal of this assignment is to implement and understand several key machine learning and deep learning algorithms from scratch, including k-NN, softmax classifiers, and neural networks, through practical coding exercises. This will deepen your grasp of both theory and application.

First setup guide - https://www.youtube.com/watch?v=xPyNYUK-3HA

Question 1: k-Nearest Neighbor classifier

For this task, please follow the steps outlined in the file knn.ipynb. This task includes filling in the code inside:

knn.ipynb (code and inline questions)

k_nearest_neighbor.py

Note: with the free resources plan the code should run in about 280 seconds for one or two loop classifiers.

Question 2: Implement a Softmax classifier

For this task, please follow the steps outlined in the file softmax.ipynb. This task includes filling in the code inside:
cs231n/classifiers/softmax.py

cs231n/classifiers/linear_classifier.py

and also, code and inline questions inside softmax.ipynb

Note: In softmax.pynb the learning block can take up to 10 minutes of running time.

Question 3: Two-Layer Neural Network

For this task, please follow the steps outlined in the file two_layer_net.ipynb. This task includes filling in the code inside:

cs231n/layers.py

cs231n/classifiers/fc_net.py

and also, code and inline questions inside two_layer_net.ipynb

Note: Total running time is about 16 minutes.

Question 4: Higher Level Representations: Image Features

For this task, please follow the steps outlined in the file features.ipynb.

Note: Total running time is less than 3 minutes and should be mainly on extracting features.

Question 5: Training a fully connected network

For this task, please follow the steps outlined in the file FullyConnectedNets.ipynb. In addition, fill the code inside the file cs231n/optim.py.

Note: Total running time should be about 45 minutes.

Important Note:

The following IPython magic commands are intended for local Jupyter environments to auto-reload changed modules:

%load_ext autoreload

%autoreload 2

In Google Colab this autoreload extension may not be available or compatible, which can cause errors when running the notebook. **Please delete or comment out** the two lines above so the code will run properly in Colab.

Required Submission Files:

Prepare the following files:

- a1_code_submission.zip: Generate by running collect_submission.ipynb inside the assignment1 directory.
- Simpler .txt file named [Your_Name_ID].txt with a link to your Github project. Make sure your link is publicly available.

Submit the following files (generated by collect_submission.ipynb) after you finish running them:
1. To the VPL (Assignment 1 - Automated testing):

1. features.ipynb
2. FullyConnectedNets.ipynb
3. knn.ipynb
4. softmax.ipynb
5. two_layer_net.ipynb

2. To Assignment 1 – submit only the txt file with your Github link.

In addition, you should extract all the files inside a1_code_submission.zip and upload them into your Github project. A link into your Github project should be in collect_submission.ipynb and also in a simpler .txt file named [Your_Name_ID].txt.

The submission must include written answers (inside ipynb files), all required uploaded files, and the corresponding code, so that every component of the assignment can be reviewed and fully graded.

This assignment must be submitted individually.

VPL Submission System

Submit your assignment through the VPL module in Moodle (Lambda), our dedicated platform for programming tasks.

Academic Integrity Checks

VPL includes a built-in plagiarism detection tool that compares submissions within the course—and optionally across prior semesters—to identify similarities and ensure originality.

Instructors review flagged cases manually for fair assessment, upholding academic honesty standards.