

Guidelines

For this assignment, you will perform binary classification (happy/sad) on facial expressions using a CNN on the same face images we used in the last homework. The main goal is for you to use a deep learning library of your choice to design and run a 2-layer CNN on the images. Although the original dataset contained 7 emotional classes, similar to Homework 3, we will only use two of them (happy and sad) for this assignment.

You can download the data from myCourses **Contents**→**Programming assignments**→**Homework3** if you did not save a copy from the last homework. Feel free to use any library of your choice to train your 2-layer CNN on `fer3and4train.csv`. And as we did before, validate your network using the dataset `fer3and4test.csv`.

Feel free to go on the internet to familiarize yourself with one of the popular deep learning libraries. I would strongly recommend **PyTorch** as its learning curve is less steep than some of the others, but it is not a requirement for this assignment. Below is a tutorial that can be valuable for building your CNN if you are new to deep learning: <https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/>

Feel free to check out other tutorials or implementations online but ensure you state them in your report.

Requirements

You should perform this assignment using Python along with any deep learning CUDA-based library of your choice. It is due on **Friday December 13th by 11:59pm**. You are required to submit your code in a Jupyter notebook along with a brief report containing short write-ups from the question(s) in the assignment. Your PDF and IPYB files should be in a zipped file and uploaded to myCourses via Assignments (formerly known as Dropbox) before the due date.

Your submitted zipped file for this assignment should be named **LastnameFirstname_hw4.zip** and should contain at least two files - *LastnameFirstname_hw4.pdf* and *LastnameFirstname_hw4.ipynb*. Feel free to submit any other auxiliary files required to run your code. We should be able to execute your code for the assignment from your submitted Jupyter notebook. Include a **Readme** file if necessary.

Problem 1. Two-layer CNN implementation

Write your own 2-layer CNN to take in the input training data from the file provided in your last homework 3. Once training is completed, test your results on the validation set provided. In your PDF report, please address out the following:

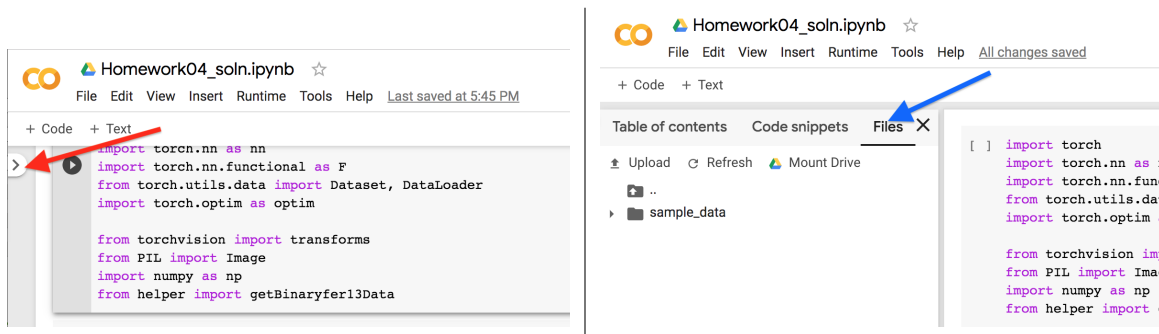
- i State your network parameters, such as your batch size, number of epochs trained for and learning rate
- ii Provide the accuracy scores from (a) training and (b) testing (or really validation) . If they are significantly different, explain briefly why this is the case.
- iii Lastly, compare the numbers you are getting for this exercise with the results from the networks you wrote previously in Homework 3 on the same exact datasets.

Using Google Colab

Google Colab will allow you to use high-end resources including access to a GPU for free, for up to 12 hours, at which time the system will refresh and you will lose any code or data not stored in another location. Ensure you save constantly and download your notebook often to avoid loss of data.

To access Colab, follow this:

- Log on to Google with your personal gmail id.
- Go to <https://colab.research.google.com/>
- From the Menu, select File→Upload notebook and upload Homework4.ipynb
- Next, from the Menu, select Edit→Notebook settings and check that the Runtime type is Python 3 and select Hardware accelerator to GPU.
- Click on the arrow at the left of the window (see red arrow) as shown in the image below, to expand the tab as shown on the right image.



- Select the sub-menu on the tab named Files (blue arrow on right image); note that this is different from the high-level option.
- You can drag-and-drop `helper.py` and the data file to make them accessible to the notebook. Uploading the data file is VERY SLOW; You might want to save it in your google drive and simply mount when needed.