CARNEGIE MELLON UNIVERSITY

10601A/C-F18: Homework #7 - "Finetuing Convolutional Neural Networks"

TA: Qiqi Xiao (qiqix@andrew.cmu.edu)

Qiqi's Office Hours:

Sat. 11/3	3pm-5pm	GHC 5 Commons near 5508
Sun. 11/4	3pm-5pm	GHC 5 Commons near 5508
Mon. 11/5	3pm-5pm	GHC 5 Commons near 5508
Tue. 11/6	6:30pm-8:30pm	GHC 5 Commons near 5508
Wed. 11/7	6:30pm-8:30pm	GHC 5 Commons near 5508
Thu. 11/8	6:30pm-8:30pm	GHC 5 Commons near 5508

Assigned: Friday, 2 November 2018.
Due: 11:59:59pm on Thursday, 8 November 2018.
You are allowed up to 10 submissions.
Late Penalty: 20% per day.

Course Policies

Previously Used Assignments

Some of the homework assignments used in this class may have been used in prior versions of this class, or in classes at other institutions, or elsewhere. Avoiding the use of heavily tested assignments will detract from the main purpose of these assignments, which is to reinforce the material and stimulate thinking. Because some of these assignments may have been used before, solutions to them may be, or may have been, available online, or from other people or sources. **It is explicitly forbidden to use any such sources, or to consult people who have solved these problems before. It is explicitly forbidden to search for these problems or their solutions on the internet.** You must solve the homework assignments completely on your own. **For programming assignments, this means you must write your programs completely by yourself, and not use any code from any source whatsoever.** I will be actively monitoring your compliance, and any violation will be dealt with harshly. Collaboration with other students who are currently taking the class is allowed, but only under the conditions stated below.

Collaboration Among Students

The purpose of student collaboration is to facilitate learning, not to circumvent it. Studying the material in groups is strongly encouraged. It is also allowed to seek help from other students in understanding the material needed to solve a particular homework problem, provided no written notes are shared, or are taken at that time, and provided learning is facilitated, not circumvented. **The actual solution must be done by each student alone.**

The purpose of programming assignments in this course is to make sure you truly understand the relevant techniques. In my more than 20 years of teaching, I have found no better way to achieve this than by having each student struggle by him/herself to implement these techniques "from scratch". For this reason, in the case of programming assignments **all code must be written by each student alone.** We will strictly enforce this policy, by carefully inspecting your code using sophisticated detection techniques. You have been warned!

The presence or absence of any form of help or collaboration, whether given or received, must be explicitly stated and disclosed in full by all involved. Specifically, each assignment solution must include answering the following questions:

- Did you receive any help whatsoever from anyone in solving this assignment? (Yes / No). If you answered 'yes', give full details (e.g. "Jane Doe explained to me what is asked in Question 3.4").
- Did you give any help whatsoever to anyone in solving this assignment? (Yes / No). If you answered 'yes', give full details (e.g. "I pointed Joe Smith to section 2.3 since he didn't know how to proceed with Question 2").
- Did you find or come across code that implements any part of this assignment? (Yes / No) (See below policy on "found code"). If you answered 'yes', give full details (book & page, URL & location within the page, etc.).

If you gave help after turning in your own assignment and/or after answering the questions above, you must update your answers before the assignment's deadline, if necessary by emailing the TA in

charge of the assignment.

Collaboration without full disclosure will be handled severely, in compliance with CMU's Policy on Cheating and Plagiarism.

Policy Regarding "Found Code"

You are encouraged to read books and other instructional materials, both online and offline, to help you understand the concepts and algorithms taught in class. These materials may contain example code or pseudo code, which may help you better understand an algorithm or an implementation detail. However, when you implement your own solution to an assignment, you must put all materials aside, and write your code **completely on your own, starting "from scratch"**. Specifically, you may not use any code you found or came across. **If you find or come across code that implements any part of your assignment, you must disclose this fact in your collaboration statement even if you didn't use it.**

Duty to Protect One's Work

Students are responsible for pro-actively protecting their work from copying and misuse by other students. If a student's work is copied by another student, the original author is also considered to be at fault and in gross violation of the course policies. It does not matter whether the author allowed the work to be copied or was merely negligent in preventing it from being copied. When overlapping work is submitted by different students, **both students will be punished.**

To protect future students, do not post your solutions publicly, neither during the course nor afterwards.

Severe Punishment of Violations of Course Policies

All violations (even first one) of course policies will **always** be reported to the university authorities, will carry **severe** penalties, usually **failure** in the course, and can even lead to **dismissal** from the university. This is not an idle threat–it is my standard practice. You have been warned!

0 DESCRIPTION

The goal of this assignment is for you to be familiarized with deep learning. Specifically, you need to learn fine-tuning Convolutional Neural Networks.

The programs you write will be automatically graded using the CMU Autolab system. You may write your programs in **Python**, **Java**, **C**, or **C++**. However, **python is strongly recommended** because of rich support of deep learning frameworks that ease the training process. You should use the same language for all parts below.

Download from autolab the tar file ("Download handout"). The tar file will contain all the data that you will need in order to complete this assignment. In addition, you will also need to create the following files before you submit (see the sections below for more details):

- `finetune.{py|java|c|cpp}`
- `submission.txt`
- `collaboration.txt`

Do not modify the structure of the directory or rename the files therein. Answer the questions below by completing the corresponding file(s), and then compress all files listed above into a `.tgz` file containing your source code and the collaboration file.

You can create this archive by running the following command:

```
$ tar -cvf hw7.tgz *.{py|java|c|cpp} *.txt
```

DO NOT put the above files in a folder and then tar gzip that folder. You must compress the files directly into a `.tgz` file and submit it to the Autolab online.

You are allowed a maximum of 10 submissions until the deadline (see front page of this handout). **If you need an extension, please contact the TA-in-charge as soon as you are aware of such need and provide your reason. Do not expect to get extensions one day before the deadline because you started homework just then and realized that it is impossible to finish it by the deadline. Please plan to spend the FULL WEEK to work on this homework. It will be really time consuming if you are new to the material. YOU HAVE BEEN WARNED!**

Besides this writeup, you are also provided with a handout tarball “hw7data.tar” containing all the data you need. To untar the file, use command

```
$ tar -xvf hw7data.tar
```

1 TRANSFER LEARNING BY FINE-TUNING [100 POINTS]

1.1 GENERAL INSTRUCTIONS

The goal of this section is to familiarize you with deep learning, by ***implementing transfer learning on Google Landmark Recognition Challenge***.

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in performance that they provide on related problems.

1.2 DATASET

The dataset is a subset of the Google Landmark Recognition Challenge. But instead of thousands of landmarks, you will be focusing on only 10 locations.

1. St. Stephan's Cathedral, Austria
2. Teide, Spain
3. Tallinn, Estonia
4. Brugge, Belgium
5. Montreal, Canada
6. Itsukushima Shrine, Japan
7. Shanghai, China
8. Brisbane, Australia
9. Edinburgh, Scotland
10. Stockholm, Sweden

1.3 RESOURCES

You can use TensorFlow, PyTorch or any other deep learning framework to complete this assignment.

Here are some resources for you to refer to:

- **Installation**
  If you want to install PyTorch, visit : `https://pytorch.org/get-started/locally/`

  If you want to install TensorFlow, visit : `https://www.tensorflow.org/install/`

- **Pytorch Tutorial**
  If you want to learn Pytorch, visit: `https://pytorch.org/tutorials/`

  A good document about transfer learning in PyTorch:

https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

- **Tensorflow Tutorial**
  If you want to learn Tensorflow, visit: https://www.tensorflow.org/tutorials/

  You can either download and look at official documentation on the TensorFlow website on how to do this, or refer this neat example of how to do it in a similar manner to PyTorch.

  https://github.com/taehoonlee/tensornets

  A really neat blogpost explaining how to download weights and use them in TensorFlow.

  https://towardsdatascience.com/transfer-learning-in-tensorflow-9e4f7eae3bb4

- **Load Data**
  First thing if you are using any of the platforms is to make sure you can iterate through the data.

  For Pytorch, visit https://pytorch.org/tutorials/beginner/data_loading_tutorial.html

  For Tensorflow, it is a little bit more complicated.
  You can visit https://www.tensorflow.org/guide/datasets

Note 1: Different types of networks assume different input image shapes, make sure you resize the images before using the images as input.

Note 2: Depending on your expertise with the Deep Learning Platforms, this exercise might take a while to do. So please start early!

Note 3: You should not be needing heavy computational resources for this homework. A good result can be achieved by training a model for 5-6 hours with only 2 epoches. A baseline result of 80% was achieved by training a model using ResNet18 for 2 hours. Depending on your choice of pre-trained network, the training time can vary. So make sure you do not choose a very large network to train unless you have the time to train it!

1.4 Input/Output Specification

File descriptions:

- train.csv - The training file. Ignore all columns instead of id and landmark_id. id maps to image_name.jpg in the images folder
- test.csv - The test file for which you will be making predictions. Ignore all columns except id

Data fields:

- id - an anonymous id unique to a given image in the images folder
- landmark_id - the id of a landmark

Put your neural network's prediction result on test dataset into submission.txt.

The file should contain a header and have the following format:

```
landmark_id
0
```

```
...
5
3
1
etc.
```

You can infer the output structure from provided sampleSubmission.txt.

Here is the breakdown of the grading criteria:

Points	Description
1	collaboration.txt
50	implementation of transfer learning.
50	satisfying results on test dataset.

Good Luck! Have Fun!

2 SOME LOGISTICAL INFORMATION

Please ensure you have completed the following files for submission and follow the instructions described in the "General Instructions" section to submit a `.tgz` file containing your source code and the collaboration file:

```
finetune.{py|java|c|cpp}
submission.txt
collaboration.txt
```

Note: Please make sure the programming language that you use is consistent within this assignment

Beware! You are allowed to run your code on the autograder only 10 times. Hence, work with the development set and test your performance on autolab only when you are very confident!