CSE 398/498 - Assignment 3

Deep Learning for Computer Vision

Logistics

The submission should be made on Course site in the form of a zip file named as **<your_full_name>_cv_24.zip** by **March 25, 2024 11:59pm (EST).**

Contents of the compressed folder should be organized in the following manner:

- **data/** - folder containing input images
- **output/** - folder containing output or any helpful intermediary images
- **src/** - containing your python scripts, comment your code well
- **readme.md/readme.txt** - containing verbose description of the steps required to run your code and other observations/comments you may have for your submission. Treat this as a short report. Include any links you may have referred to in this document.
- **environment.yml** - exported conda environment file. Contains information about libraries and their versions.

Credit: 6% of course total (Start this assignment early as processing will take time!!!)

Due date: March 25, 2024, 11:59 pm

Pre-assignment

Prepare your laptop. Virtual environments are a great way to ensure stability of your code execution and also improve its reproducibility. I suggest everyone to use the **Anaconda** package manager for your projects and assignments. If you have used it before, you already know. If you haven't used it before, it will make installations easier and project management better. Also, a lot of other people's code uses it so it's good to learn.

Quick tutorial for conda: https://conda.io/projects/conda/en/latest/user-guide/getting-started.html

By now, you all should be able to provide an environment that I can reproduce. Your assignment will not be evaluated if I cannot run it. Please contact me before submitting if you have any questions or issues.

We are going to need Python3 (for instance 3.6.x) and a few libraries installed, in addition to the OpenCV library. **Please do not use Python 3.7.x for this practical**.

1. If you don't have Python3 and OpenCV, you can use your preferred way to install them, for instance install Anaconda Python (https://www.anaconda.com/download/), and then use its package manager to install OpenCV:

```
$ conda install -c conda-forge opencv
```

2. Now, all you need to do is to install `scikit-learn`, `tensorflow` and `keras`. All of this will be inside your environment. This can be done with these commands:

If you are using pip:

```
$ pip install numpy scipy scikit-learn
$ pip install matplotlib
$ pip install h5py
$ pip install tensorflow
$ pip install keras
```

If you are using anaconda:

```
$ conda install -c anaconda numpy
$ conda install -c anaconda scipy
$ conda install -c anaconda scikit-learn
$ conda install -c anaconda h5py
$ conda install -c anaconda matplotlib
$ conda install -c conda-forge tensorflow
$ conda install -c conda-forge keras
```

If you encountered any problems during these steps, these sites can help:
http://scikit-learn.org/stable/install.html
https://www.tensorflow.org/install/
https://keras.io/#installation

You may need to use slightly older versions of tensorflow and keras. So, try installing an older version if the code doesn't work with the default version. In pip, you can specify the version like this: `pip install tensorflow==2.3.1` (or another version number) whereas for conda use single = like: `conda install matplotlib=1.4.3` (or another version number)

3. Before using `keras`, we have to make sure it is set up to use `tensorflow` as the backend. Follow these instructions to make sure it is correctly configured: https://keras.io/getting_started/

4. Test the `keras` installation by calling Python and importing the `keras` module:

```
$ python
```

```
>>> import keras
```

You should see a message saying that `keras` is using `tensorflow` backend.

5. Run the example code:

```
$ python main.py --classifier resnet
```

The first time the program runs, it will automatically download the trained weights for ResNet Convolutional Neural Network [https://arxiv.org/abs/1512.03385]. This program will use a stream of video from your webcam as the input and use ResNet to classify objects in the image.

You should be able to see the top 3 classes and the confidence associated with them in upper left corner. For this example, they are:

```
Computer keyboard: 0.966
Spade bar: 0.007
Laptop: 0.007
```

Here you can check the names and images of 1000 classes which the ResNet and VGG were trained for:

http://image-net.org/challenges/LSVRC/2015/browse-synsets

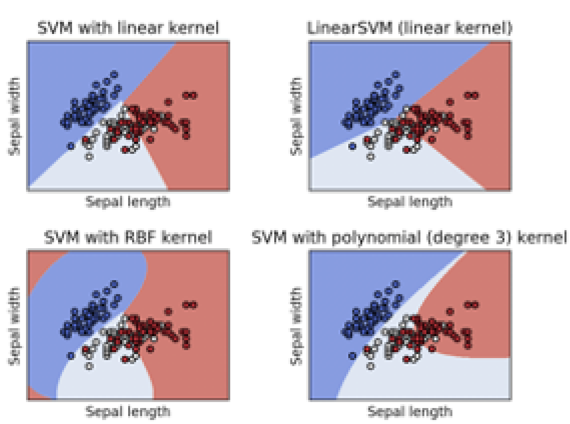
If you run the program using `--classifier vgg`, the VGG Convolutional Neural Network [https://arxiv.org/abs/1409.1556] will be downloaded and used:

6. Alternatively, you can run the program without any parameters, and press 'r' or 'v' to capture a frame and classify it using respectively ResNet or VGG.

7. Just to double check that the `scikit-learn` and matplotlib are up and running, run this example code that illustrates the SVM-based multiclass classification on a very simple dataset of flower features:

```
$ python svm_plot_iris.py
```

If everything works well, you should see the following picture:



Trouble shooting:

If you are using MacOS and encounter problems with matplotlib backend when running this example, you can find the code: https://scikit-learn.org/0.18/auto_examples/svm/plot_iris.html Or use the settings you used for matplotlib in your code for the previous assignments.

Task 1 (1 point):

- Select three classes corresponding to the objects from the xlsx file (good to select objects you can easily find at home)
- Train linear SVM using fc1 CNN codes and try to recognize all your objects online presenting them to the camera. Show output for top 3 predicted classes for each of these objects in your report.

Task 2 (1 point):

- Experiment with the level of the CNN codes (fc1 vs fc2) and SVM kernel ('linear', 'rbf', 'poly') for these three classes. Record performance with each (Take screenshots where the configuration works and where it doesn't).
- Which configuration is the best? How would you explain why this selected configuration works best?

Task 3 (1 point):

- For the best configuration found in task 2, train your SVM for the extended set of CALTECH classes (10 or more, including your 3 classes selected earlier).
- Does your solution still perform well? Show output frames with predictions in your report. What is the reason for a different performance?

In the first three tasks, you experienced extracting features from pre-trained deep learning models and training your own SVM classifier for final decision task.

In the next task, you will experience a different deep learning framework - **Pytorch**. Here you will train your own network to classify images of 10 object classes. An example network architecture that we saw in class has been provided. Please refer to the 60-minute tutorial or directly the pytorch documentation for more information about layers and their parameters.

Task 4 (3 point):

Prep: In your current environment, install pytorch

`pip install torch torchvision` or `conda install -c pytorch pytorch`

1. Download the CIFAR-10 dataset from http://www.cs.toronto.edu/~kriz/cifar.html. Make sure you read how the dataset has been formatted.

2. Untar the dataset and place it in the data folder as we usually have done for our other assignments. Make sure you update the path in line 9 to correspond to your data. (Do **not** upload this folder when submitting the assignment to Course site)
3. Load the CIFAR-10 dataset, a dataset with 50 thousand training images of size 3x32x32 each belonging to one out of 10 possible categories of objects. An example provided in the task4.py file.
4. Visualize some of your images to understand the dataset and the nature of images. An example provided in the task4.py file. Change indices and view other images.
5. You will now define a convolutional neural network classifier and compute the accuracy of the model on CIFAR-10. Feel free to create any network that you consider would work well. You can get inspiration in some of the convolutional neural network architectures studied in class. I recommend you start with the LeNet architecture included in task4.py and start from there.
6. Train your proposed CNN model and compute the accuracy.
7. **(For CSE 498 students only)** Plot the loss for training and validation as training progresses, and the accuracy for training and validation. Use matplotlib to plot two linecharts, one should have the training accuracy and the validation accuracy in the same figure, the second figure should have the training loss and validation loss in the same figure. The x-axis should indicate the epochs, the y-axis should indicate a range of values for either the accuracy (first figure) or the loss (second figure).
What is the final accuracy?

General Hints: Remember to use classifier.train() classifier.eval() for the two different classifier modes, also remember to normalize the input data.

Rubric

Task 1 : 1 point for correct implementation and displaying output.
Task 2 : 1 point for correct implementation of different possibilities and explanation.
Task 3 : 1 point for correct training with more classes and explanation of performance.
Task 4 : 1 point for step 1-5 where we define model architecture and prepare data.
For CSE 398, 2 points for the training code and accuracy calculation on validation set.
For CSE 498, 1 point for the training code and 1 point for analysis with the help of the plots.

<u>Deadline Extensions and Late Submissions:</u>

- First deadline extension request will be granted with a 3-day extension automatically and you won't be penalized. ***Applicable to Assignments and Projects***
- For subsequent late submissions, you will lose 10% for each day late for the programming projects. This means anytime within the first 24 hours after the due date count as 1 full day, up to 48 hours is two and 72 for the third late day. Beyond that, your submission will not be graded. ***Applicable to only Assignments***

Please clarify this with the instructor if the policy is not clear before you submit late.

Statement on Academic Integrity

University -We, the Lehigh University Student Senate, as the standing representative body of all undergraduates, reaffirm the duty and obligation of students to meet and uphold the highest principles and values of personal, moral and ethical conduct. As partners in our educational community, both students and faculty share the responsibility for promoting and helping to ensure an environment of academic integrity. As such, each student is expected to complete all academic course work in accordance to the standards set forth by the faculty and in compliance with the University's Code of Conduct.

Course - The work you do in this course must be your own. This means that you must be aware when you are building on someone else's ideas—including the ideas of your classmates, your professor, and the authors you read—and explicitly acknowledge when you are doing so. Feel free to build on, react to, criticize, and analyze the ideas of others but, when you do, make it known whose ideas you are working with. If you ever have questions about drawing the line between others' work and your own, ask me and I will give you clear guidance or you may visit Lehigh Library's 'Proper Use of Information' page at http://libraryguides.lehigh.edu/plagiarism

Grade Specific - Zero assigned to the Quiz/Assignment for first offense and the student will Fail the class on second offense.
For assignments, you can discuss with peers but the code should be your own. For quizzes, no consultation with a living or non-living entity is allowed.

University COVID Policy:
To meet the challenge of teaching and learning during the COVID-19 pandemic, Lehigh instructors and students will be adopting new forms of instruction and interaction; following new guidelines around classroom behaviors; enhancing communications; and doing our best to be patient, flexible, and accommodating with each other. In remote synchronous meetings, students are expected to attend just as they would any other Lehigh class. Zoom classes work best when all students come to class ready to participate and follow the instructor's guidelines regarding use of web-cameras. You may be asked to turn your camera on during active learning sessions in Zoom. If you have a strong preference not to do so, please contact your instructor to let them know. Students should respect the in-classroom privacy of their instructors and fellow students by not taking screenshots or recording class sessions. Some instructors will record Zoom sessions; however, any recorded live sessions will be shared only with students in the class and will be deleted at the end of the semester.

In our physical classrooms, Lehigh has established a policy requiring everyone to wear face coverings when in public spaces inside buildings on our campus and to maintain social distance. This policy applies to our physical classroom. Thank you in advance for following this rule. Students who do not wear a face covering during in-class meetings will be reminded to put their face covering on. If they do not do so, they will be asked once again to do so or leave the classroom.