Deep Learning
Exam #2
Amir Jafari

1 Introduction

In this exam, we will be using the original dataset we used to get the dataset for Exam 1, which consists of images with more than one class on them, each of them labeled by encoded categories. The goal is to use a Neural Network to tell all the types of images present in each frame, which is a Multi-Label classification problem. In order to train your model, you will be given an excel file with a training set and a test set. Your goal is to output the correct one-hot-encoded target (see below).

The format will be a competition, exactly the same one used for Exam 1. Toward the leaderboard, we will use the Binary Cross-Entropy loss. The targets will be one-hot-encoded following the order: ['Class1', 'Class2', 'Class3', 'Class4', 'Class5','Class6','Class7','Class8','Class9','Class10']. For example, if an image contains Class1 and Class3 types, the target will be [1, 0, 1, 0, 0, 0, 0 ,0 ,0 ,0].

Note that the data categories differ from Exam1, so your model will be even worse if you add the Exam1 data for training.

2 Dataset and Sample code

- Download the exam dataset and sample code from the following link https://storage.googleapis.com/exam-deep-learning/Exam2-v4.zip.
- You can use this link and use `wget` command to download your data into your cloud. For example

```
- wget https://storage.googleapis.com/exam-deep-learning/Exam2-v4.zip
- wget -O Exam2-v4.zip https://gwu.box.com/shared/static/ohyhvlz65e2x7t9ktqto5fmownjc8q2j.zip
```

- unzip it and you should have Exam2 folder that includes **Code, Data and excel** directories.

```
- unzip Exam2-v4.zip
```

- Please check the sample codes first and download them from the downloaded file and then you can write your own codes. This is just to get you started and give you an example of how you write your predict function and save your models. You will get a sample directory

  1. **Code** : the example code
  2. **Data** : images example
  3. **excel** : excel example

- The datasets consist of an Excel file and the images to be tested. The name of the images are listed in the spreadsheet classified with “train” and “test” labels. Use the “train” to create your model and “test” to tested. A very important note, the “test” images should be used as they are label do not change this parameter since these exact images will be used to compare the validity of the models. Only the valid models using this exact information will be given a score in the leaderboard.
- Note: Use pandas to read the dataset and work on DataFrames.

3 Rules of Competition

Please read these rules **carefully** and if you have any questions please send an email to me directly.

- You can use Convolutional Layers however feedbacks are not allowed (LSTM or any RNN architecture). Pre-trained models are allowed such as (resnet, densenet, ...)
- You can use other operations in-between layers, like Dropout, Batch Normalization or other types of layer input/output manipulation.
- You can only use the data you are given. Using additional data from any other sources is not allowed.
- You can do any kind of pre-processing with the training data (the data is already splitted in the excel file)
- You can only use PyTorch for training the model.
- Every student will have a nickname for the competition. You are not allowed to share your nickname with anyone or reveal other students’ nicknames. If we find out you will get **zero** grade for the Exam.
- You are not allowed to copy code or ideas from any students in the class. If we find out you will get **zero** grade for the Exam.
- You are allowed to search in the internet and find out ideas. You can use any external GitHub but you need to **cite** it. If we found any violation of this rule you get a reduce grade.

4 Daily Deliverables

1. A single script named `test_<Nickname>.py`. You should receive your nicknames by email, if you do not, please contact us. (e.g. test_katrina.py)
2. A single model `model_<Nickname>.pt` .
3. A file text `summary_<Nickname>.txt`, which contains model.summary().
4. An Excel file `results_<Nickname>.xlsx`, with the following format

- “id” : image name
- “target”: original classification
- “split” : test images, this images are given in the original Excel.
- “results” : the results of the prediction

5. The names should be as listed above.
6. You need to submit item 1 through item 4 every single night to Blackboard (in electronic reserve section - there is a submission link for each day) if you want to test your trained model for that day. The results of daily submissions will be posted the next day in Daily Board excel sheet in the electronic reserve section of BB.
7. **Very Important Note**: DO NOT ZIP YOUR FILES IF YOU DO THEN YOU DON’T GET A GRADE FOR THAT DAY. PLEASE UPLOAD THEM SEPARATELY AS THEY ARE.
8. **Note**: You **just** have **one** submission per day. In case of ensembling you need to write your predict function in a way that it loads multiple models and does the ensemble and then predict. Makes sure all of your models named properly (i.e, `model_<Nickname>1.pt`, `model_<Nickname>2.pt`, etc.) and you need to put all of your models in **ONE SUBMISSION**. In other words, one submission per a day means you just get one score per a day for your trained model.If you do not submit your models for one day or so you do not get penalized.

5 Clarifications regarding test_<nickname>.py

- The python script must begin with the necessary imports, followed by one or more lines of `os.system("sudo pip install <whatever additional package you used>")`. Please make sure this way of installation works on the ubuntu image. If not, write whatever command line commands you need using `os.system`.
- The rest of the code must consist of a single function called test with the following features:

  a-1) The test should be run in the terminal with two arguments `--path --split`.

  a-2) python3 test_Nickname.py –-path ’path to your exam directory’ –-split test

  a-3) Example: python3 test_adam.py –path /home/ubuntu/Exam2 –split test

  a-4)) The excel file will obtained from `path + os.path.sep + "excel"` , where the excel file with the name of ids and the targes are recorded.

- The “split” arguments can receive two values : “test” and “validate.” The “validate” will use data that we have in store to calculate the metrics for the leaderboard.
- The test script most read the images from the path given in the previous argument.
- The test script may have any transformation needed to make your model works correctly.

- The script must generate a excel file with results of your prediction in the format already mentioned in section 4 of this document.
- The script must be save the model that is being loaded as `'summary_.txt'.format(NICKNAME)`. An example is provided in the training example and the `test_to_do.py`
- The script must save the results into an .xlsx file. The format is explained in section 4 of this document. Remember to use exactly the same format.
- The test script will be using the models ".pt" in the local directory.
- Test the file in your linux instance. Only send test files that run without errors.
- Note that the process to get the leaderboard results will be fully automated, so if Python throws an error, your model will be skipped and you will see a score of 0 and the corresponding error message instead of your model's score. You can contact me anytime to fix this if you think your code has no bugs.

6 Final Deliverables - Last day of Exam

1. Create a folder named `Final_Scripts`. Put the following into it.
   - A final script named `train_<nickname>.py` that shows everything you did to get your best model.
   - A final test file like the one you submitted during the competition.
   - Your best model.
2. Create a folder named `All_Submissions`. Put the following into it.
   - All daily train, test scripts and models for each day. For example, for day 1, you need to have a subfolder in the `All_Submissions` folder named as `day1` with all the scripts and models of day1 in it. Do the same for the rest of the days.
3. Create a folder named `Exam_report`. Put the following into it.
   - An exam report named `report_<nickname>.pdf` that explains what you did and why (like changes that made your score higher, creative ideas, etc). Please submit in `.pdf` format.
   - Your report should not be very lengthy or very short. It needs to show your work It can have the followings (network architecture, table, block diagram,what changes you did, preprocessing, new ideas, tricks, snippet of your code ...)

Your `train_<nickname>.py` scripts and your `report_<nickname>.pdf` report will only be graded after the competition ends, so you don't need to submit them until then.

There is a link final submission for exam2 in electronic reserve section of BB. Please submit your final submissions to this link. Please also submit a copy of the pdf file to BB (file upload).