Instructor: Lei Li UCSB CS 190I Deep Learning

Machine Problem 1

January 17, 2023

1 Background

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. In this homework, you will train a CNN-based model for binary classification on the CelebA dataset, make predictions on unseen images, and submit your predictions.

We will use Kaggle to host the competition. A baseline template is provided, and you need to make improvements of the provided baseline to achieve a higher accuracy on the test set.

2 Preparation

2.1 Register and Join the Competition.

Go to Kaggle and register an account **using your UCSB email**. Join the competition through this CS190 Winter23 Deep Learning MP1.

2.2 Competition on Kaggle

1. Register in Kaggle and go to CS190 Winter23 Deep Learning MP1.
2. Read the *Overview* and *Data* part.
3. Choose the *Code* tab and click **New Notebook** to create a notebook for this competition. The competition data will be automatic added under the input directory of the workspace of this notebook.
4. Use the **Add Data** on the right and search CelebFaces Attributes (CelebA) Dataset. Click the '+' button to add it to the current workspace.
5. Follow the template script provided under the *Overview* tab or write your own code. Train the model and predict the results on the test set.
6. When you finish, submit the *submission.csv* to the competition. You can see your performance under the *Leaderboard* tab in a few minutes. Also remember to submit your notebook.

2.3 Computation Resource

There are two different options to develop your model. Kaggle provides free computing resources including GPU machines with its Notebook feature. You could also choose to download the python script and develop on your local computer. We also provide computation resource from ECE Machine Learning Lab.

2.3.1 Kaggle Notebook

1. Open your nodebook and click the three dots on the right side of editor bar.
2. Click **Accelerator** and choose the GPU you want. The default is None.

2.3.2 Local Machine

1. Download the dataset into a directory (e.g. *cs190i-mp1*) on your machine: *celeba-dataset*, *cs190-winter23-deep-learning-mp1*.
2. Download and install Anaconda following the instructions.
3. Create the development python environment with the provided file and activate the environment.

```
conda env create -f environment.yml
conda activate mldl
```

4. Under your working directory, use the command **jupyter notebook** to start the jupyter service. It will automatically open your browser and list your files under the working directory. Create a new notebook or open the existing notebook you want to start.

2.3.3 ECE Machine Learning Lab

We create an account for each student in ECE lab. Here is some information about it:

- Operating System: Ubuntu 20.04
- Hardware: Total of 9 systems
- CPU: Intel i9-10900x
- Memory: 64GB
- GPUs: (2) Nvidia RTX A4000 GPUs
- 5TB of storage available for datasets mounted on */mldata*
- Installed Software :
  - python 3.8.10
  - pycharm

- miniconda
- Visual studio code
- Nvidia-460-cuda
- Git

Check Machine Learning Lab Status webpage (contains a link to usage instructions) and Instructions to access the lab for more information.

To access the servers:

1. First, you need to change the default password. The instruction can be found here.
2. After resetting the password, you can ssh to the server by your account and password. The server ip address is *mllab01.ece.ucsb.edu* to *mllab09.ece.ucsb.edu*. For example, you may ssh to the service by:

   ```
   ssh youraccount@mllab01.ece.ucsb.edu
   ```

   and input your password. The account name and password are the same for all servers.

The rest is the same as Section 2.3.2.

3 Start Coding

We provide a *template.ipynb* and *template.py* for you to start with. If you are familiar with the project, you can also write your own code from scratch.

If you want to use the template, make sure you read the code and comments and understand what each part is doing. You will need to complete the **TODO** to make the code run.

It is optional to use the template. If you do not need it, you can just write your code and submit it, ignoring the template.ipynb or template.py.

The template includes 6 parts:

1. CelebA Overview
2. Dataset for Smiling
3. Define the training and validation loops
4. Define the Model Structure
5. Training

6. Predict and Submit

In the forth part, we define a very simple model (MLP) to do the classification. We also provide a LeNet with some **TODO**. Feel free to replace it with any other model architecture to get a better results. You can even finetune from an existing pre-trained model.

Hint on Improving the Performance. There are mainly two aspects of the model you can choose to improve upon the baseline:

1. Model structure: use more hidden layers, try different hidden sizes, add dropout layers, etc.
2. Optimization: try different optimizers other than SGD, tune the learning rate and other hyperparameters, etc.

4 Submission and Grading

When you are satisfied with the model performance, make a submission to the competition. There is a daily limit of 20 submissions per student.

After a successful run, a CSV file named `submission.csv` containing all predictions for the test data should be written to the working folder. Go to the competition page and submit the file.

Code. In addition to making submissions on the competition website, you need to submit your code either in `.py` or `.ipynb` format to Gradescope. Your code should run as is and should produce the exact same results as your submission on Kaggle.

Grading. If the submitted code does not run out of the box, you would receive 0 pts regardless of the Kaggle competition score. Otherwise, your grade depends on the private score of your model predictions on Kaggle. The private leaderboard uses a different set of test examples to the public leaderboard and will only be revealed after the deadline. You should select one of your submissions as your final submission. The grading scheme is as following:

run and get a result lower than 90%:	40 pts
higher than 90% :	60 pts
higher than 93% :	100 pts
top 5 submissions on the private leaderboard:	+10 pts