DDA4220/MDS6224/MBI6011 Deep Learning

Assignment 1: Image classification by using deep models

Due Date: 23:59, 19th March, 2023

This assignment aims to train models for flower classification. You can choose either Colab online environment or local environment. This assignment will worth **15%** of the final grade.

Exercise 1: Fine-tune classification model using MMClassification (50%)

Please complete the fine-tune training based on the pre-training model provided by MMClassification (https://github.com/open-mmlab/mmclassification). You should:

1. Prepare the flower datasets.
   The flower pictures are provided in `flower_dataset.zip`.
   The flower dataset contains 5 categories of flowers: daisy 588, dandelion 556, rose 583, sunflower 536 and tulip 585. Please split the dataset into training set and validation set in a ratio of 8:2, and organize it into ImageNet format. Detailed steps:
   1) Put the **training set and validation set** under folders named '`train`' and '`val`'.
   2) Create and edit the **category name file**. Please write all names flower categories into file '`classes.txt`' with each line representing one class.
   3) Generate training (optional) and validation sets **annotation lists**: '`train.txt`' and '`val.txt`'. Each line should contain a filename and its corresponding annotation. Example:

```
daisy/NAME**.jpg 0
daisy/NAME**.jpg 0
...
dandelion/NAME**.jpg 1
dandelion/NAME**.jpg 1
...
rose/NAME**.jpg 2
rose/NAME**.jpg 2
...
sunflower/NAME**.jpg 3
sunflower/NAME**.jpg 3
...
tulip/NAME**.jpg 4
tulip/NAME**.jpg 4
```

   The final file structure should be:

```
flower_dataset
|--- classes.txt
|--- train.txt
|--- val.txt
|  |--- train
|  |    |--- daisy
```

```
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- dandelion
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- rose
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- sunflower
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- tulip
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |--- val
|   |    |--- daisy
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- dandelion
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- rose
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- sunflower
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
|   |    |--- tulip
|   |    |    |--- NAME1.jpg
|   |    |    |--- NAME2.jpg
|   |    |    |--- …
```

This process can be done using Python or other scripting programs. And it can be completed locally/offline to save the Colab's time online.

Once the dataset has been prepared, please migrate the processed dataset to the folder `mmclassification/data`. To reduce duplicate uploads, you can Sync the data to google drive

and import it in Colab.

2. Modify the configuration file

   Use the `_base_` inheritance mechanism to build profiles for fine-tuning, which can be inherited and modified from any ImageNet-based profile provided by MMClassification.

   1) **Modify the model configuration.** Change the category header to adapt the model to the number of data categories in our flower dataset.
   2) **Modify the dataset configuration.** Change the data paths for the training set, validation set, the list of dataset annotations, and the category name file. And modify the evaluation method to use only the top-1 classification error rate.
   3) **Modify learning rate strategy.** Fine-tuning generally uses a smaller learning rate and fewer training period. Therefore please change them in configuration file.
   4) **Configuring pre-trained models.** Please find the model file corresponding to the original configuration file from Model Zoo. Then download it to Colab or your local environment (usually in the `checkpoints` folder). Finally you need to configure the path to the pre-trained model in the configuration file.

3. Complete the finetune training using tools.

   Please use `tools/train.py` to fine-tune the model and specify the work path via the `work_dir` parameter, where the trained model will be stored.

   Tune the parameters, or use a different pre-trained model to try to get a higher classification accuracy. For reference, it is not difficult to achieve classification accuracies above 90% on this dataset.

Exercise 2: Complete the classification model training script (50%)

The provided script `main.py` is a simple PyTorch implement to classify the flower dataset you've prepared above, but this script is not complete.

1. You'll be expected to write some code in some code blocks. These are marked at the top of the block by a `#GRADED FUNCTION` comment, and you'll write your code in between the `### START SOLUTION HERE ###` and `###END SOLUTION HERE###` comments.
2. After coding your function, put your flower datasets `flower_dataset` to the `EX2` folder (`EX2/flower_dataset`) and then run this `main.py` script.
3. If your code is correct, you can obtain the right printed information with loss, learning rate and accuracy on validation set, and the best model with the highest validation accuracy will be stored in the `Ex2/work_dir` folder.
4. You can modify the configuration or the model in `main.py` to beat the original result. (optional)
5. Please write a report with Latex and submit a .pdf file. Please use this overleaf template https://www.overleaf.com/6589317849mydqrqpkfbwk . There are detailed report requirements.

Submission requirements:

1. You need to submission all materials to **GitHubClassroom**. **Please create a GitHub account in advance. (https://github.com).** Later we will provide a link of this assignment, click it and you will get an initial repository containing two folders named: `Ex1` with `flower_dataset.zip` in it, and `Ex2` with `main.py` in it. You need to upload all the materials below to your repository:

1) For **exercise 1**, please put your **configuration file** and **the saved trained model** in `Ex1`;
2) For **exercise 2,** please put your **report**, **completed script file** and the **saved trained model** (auto saved in `work_dir`) in `Ex2`.

2. Please note that, the teaching assistants may ask you to explain the meaning of the program, to ensure that the codes are indeed written by yourself. Plagiarism will not be tolerated. We may check your code.
3. The deadline is **23:59PM, 19th March**. For each day of late submission, you will lose 10% of your mark in corresponding assignment. If you submit more than three days later than the deadline, you will receive zero in this assignment. No late submission emails or message will be replied.