Fall 2023

CSC 578 HW 7: CNN Image Classification

- **Graded out of 5 points.**
- Do all questions

Overview

The objective of this assignment is to enhance your understanding of Convolutional Neural Networks (CNNs) while gaining experience with Kaggle competitions at the same time. Kaggle is an online machine learning community. It provides a large collection of public data sets and hosts competitions to solve data science and machine learning challenges.

The assignment is to experiment with CNNs and write about your experience on model building and hyperparameter tuning and the analysis of results. The CNN experimentation part is made into a **Kaggle inClass competition (https://www.kaggle.com/t/bd137405a253475c844b81cd34326d42)** so that you can compete with other students in the class.

The competition is to classify the dataset called '**Intel Image Classification**'. This Data contains a total of around 17k color images of size 150x150 pixels, distributed over 6 categories: {'buildings', 'forest', 'glacier', 'mountain', 'sea', 'street'}. In this assignment, you will use CNNs to classify the images. The dataset is already divided into training and test sets. You use the training set to train a CNN model (though you further split the training set to training and validation subsets in the code). After training the model, you generate predictions for the test set and submit the predictions to the competition.

Deadlines:

- The competition will close on **Sunday, November 12 at 5:59 pm, Central time**. There can be no submissions to Kaggle after this time.
- Homework submissions on D2L will be accepted until **11:59 PM** of the day.

Also note that this competition is intended for ***fun***. Don't be concerned or take it personally if you did not place high in the leaderboard; ranking will not affect your homework grade.

Code Development

Development Platform:

You can develop your code wherever you like, but you are advised to use a platform/environment where hardware accelerator (GPU or TPU) is available to speed up the execution time. You can also use cloud platforms, e.g. Kaggle, Google Colab and AWS, or local machines.

- If you want to use **Kaggle's cloud development tool**, you can open the 'starter' code in Kaggle and click on the button 'Copy & Edit' to get started. Then follow this instruction document. Also be sure to **set the 'Accelerator' on the right to GPU**.

- If you want to use **Google CoLab** instead, you can directly access the dataset on Kaggle while writing code in Colab. Follow this instruction. Note the name of our competition is 'csc-578-hw7-fall-2023' (NOT dogs-vs-cats). Also **change the runtime environment to use GPU or TPU**, by `Runtime > Change Runtime Type > Hardware Accelerator > GPU or TPU`.

- Alternatively, if you want to run your notebook on your local machine, you can download the data to your local computer by clicking on the 'Download All' button.

Code Writing:

For any platform you chose, you can start with the **starter code posted on the Kaggle competition site, https://www.kaggle.com/code/norikotomuro/notebookc9f2841c24** (or download a local notebook copy, html). However note that you must:

- Add your name, course/section number and assignment name at the top of the file.
- **Edit the starter code appropriately**. Add your own comments too.
- Make a clean, *presentable* code file. You should at least:
  - **Clean** the final homework submission code. **Remove** all debugging lines and extraneous output displays.
  - **Avoid long run-off lines**. Ensure all lines (code as well as comments) are less than 80 characters long.

Model Development and Hyperparameter Tuning:

The starter code includes a simple CNN model. You should experiment as many models/hyperparameters to find an optimal configuration.

IMPORTANT: You MUST NOT use a pre-trained network (such as VGG net or MobileNet) -- you must train the model from **scratch** in this assignment or competition. However Data Augmentation is allowed.

Parameters related to models include:

- number of filters.
- size of filters (e.g. 5*5, 7*7).
- number of Convolution layers.
- number and size of Fully Connected layers.
- use of Dropout layers and the percentages.
- use of regularization.
- use of BatchNormalization layers. Look at the Keras Documentation to learn about it.

Hyperparameters related to compilation include:

- learning rate
- regularization
- drop-out percent
- mini-batch size

Model Evaluation:

- Note that the code uses categorical_crossentropy as the loss function, while Kaggle evaluation uses log_loss (which is the same as 'log likelihood'). So the performance results on Kaggle may differ from what you get during training.
- Keep in mind the performance can be measured by other aspects besides the competition evaluation metric, such as *model complexity, computational speed* and *learning stability*. Consider the trade-off between those aspects vs. performance.

Kaggle Submission File:

You must add code in the starter file to generate predictions for test images. The content of the file must be formatted **exactly** to the following specifications:

- It must be a comma-separated csv file. (and the file extension must be .csv). Also there should be no spaces between values.

- The first row must be the header, then the predictions of test images should follow.

- For each prediction row, the first value is the file number, which is the file name of the test images (in particular, the integer part of the file name, e.g. "20056.jpg"). Then the probabilities for the six target categories should be written.

- Entries (after the header row) MUST be in the ascending order of the file names/numbers.

```
fnum,buildings,forest,glacier,mountain,sea,street
20056,0.00018,0.99224,0.00243,0.00080,0.00148,0.00283
20057,0.74160,0.00422,0.00391,0.00815,0.10568,0.13642
20058,0.00070,0.00206,0.74449,0.12540,0.12668,0.00064
```

Requirements and notes on the Kaggle competition:

- You will need to submit your model predictions in order to receive a score and be put on the leaderboard.
- **You MUST make at least 3 successful Kaggle submissions, with different results every submisison (REQUIRED).**
- Note that you can submit a maximum of 10 submissions per day.
- Your score in the leaderboard is based on the best two results among all submissions you've made. So if your latest submission was worse than the current best result, your score on the board will not change. The system keeps track of the running best two results.
- Note that you will see the 'public' leaderboard while the competition is open. Then after the competition closes, the 'private' leaderboard becomes available. The way it works is the test set is divided (by Kaggle) into public and private subsets (when the competition is created), and the public leaderboard shows the ranking based on the performance for the public test set. This is also to prevent overfitting to the competition ranking (during competition). The final ranking will be the one for the private test set, so you know.
- You must make a successful submission to the competition. Failure to do so will have a significant impact on the points that you can earn.

Homework Submission

Submit the following:

1. The Jupyter notebook code file and its pdf/html version. Be sure to **add your name and course/section number at the top of the code file**.
   - Show at least the best model and its full trace of training with charts. You can leave other models in the code too, but make sure you **clearly** indicate the best model. (Make it easy to find for someone who is looking at dozens of these!)
   - In general, organize the code file well, by inserting section headers and comments/mark-downs. Code with no comments will be subject to points deduction.
   - **Please note:** The code file is submitted for verification and spot-checking. Please don't assume that the grader will search for graphs in your code. If you want something seen, put it in your report.
2. Report
   - In pdf or docx.
   - Minimum 2.0 pages of text plus visuals.
   - Be sure to add your name, course/section number and the assignment name at the top of the file.
   - Also at the top of the file, write, **in bold**, your Kaggle user name (as displayed in the leaderboard) and ranking/score (public or private, or both; at the time of writing).
   - **Content** (clearly labeled, professionally organized and presented):
     1. **Description of your journey of hyperparameter tuning.**
        - Starting from the initial model, describe in depth:
          - model architectures and hyperparameters you experimented,
          - **why** you tried them (including your **expectations**),
          - and the results you got and your reaction to them. **Make a summary table (and a chart) of the models and their results**.
        - Description of the **best model** (the one that produced the best competition performance).
        - Description of one **non-best model** -- a model that caught your attention or that you found noteworthy. Discuss its performance results and say why you found it interesting.
        - Insightful comments. A large portion of the grade will be placed on this part.
     2. **Your final conclusions** on the best model (as compared to other models)
     3. **Your reaction and reflection** on this assignment overall (e.g. difficulty level, challenges you had).
   - Reports which are nicely organized and well written, with a sufficient amount of comments and presentable graphs/charts will receive a higher grade. Ones with terse, minimal content will be considered insufficient and receive a lower grade.

DO NOT ZIP YOUR FILES. SUBMIT EACH FILE SEPARATELY.

Assessment

Submissions will be scored based on:

- 1.5 points: Making sufficient Kaggle submissions, which are a good faith effort to produce good performance on the task.
- 2 points: Excellent description of best model and hyperparameter journey
- 1.5 point: Other aspects of assignment, including: non-best model description, conclusions, reactions, overall clarity, charts of results.