Judging a Book by its Cover

COL774 : Assignment 4

Sem I, 2022-23

Due Date: Monday November 28 (2022), 11:50 pm. No late submissions allowed. Total Points: 30 + 70

Notes:

- This assignment has **two parts**: non-competitive (30 marks) and competitive (70 marks)
- You should submit all your code (including any pre-processing scripts written by you)
- Do not submit the datasets
- **Include a write-up (pdf) file**, consolidated for both parts, which includes a brief description for each question explaining what you did. Include any observations or model details required by the question in this single write-up file
- You should use **Python** as your programming language and **PyTorch** as your deep learning framework
- Your code should have appropriate documentation for readability
- You will be graded based on what you have submitted as well as your ability to explain your code. Additionally, in competitive part, you will be graded based on your model performance relative to the class
- Refer to the Piazza for assignment submission instructions.
- This assignment should be done in groups of **in groups of 2**. If you are really keen on doing it **individually**, talk to us. You should carry out all the implementation by yourself (i.e., in your group).
- Since the assignment consists of a competitive part, no late submissions are allowed.
- For the competitive part, you will submit your class labels on a leaderboard, which will display your score and your ranking with respect to other submissions in the class.
- We plan to run **Moss** on the submissions. **We will also include solutions from the internet to maintain integrity.** Any cheating will result in a **zero on the assignment**, an additional penalty equivalent of the weight of the assignment and possibly much stricter penalties (including a **fail grade** and/or a **DISCO**).

1 Problem Statement

You are given a dataset consisting of book cover images and book titles. The task is to predict the book genre amongst 30 possible categories (example: Literature & Fiction, Medical Books, Romance, and Law). The task can be achieved using either the cover images, or title text, or both, in a "multimodal" fashion!

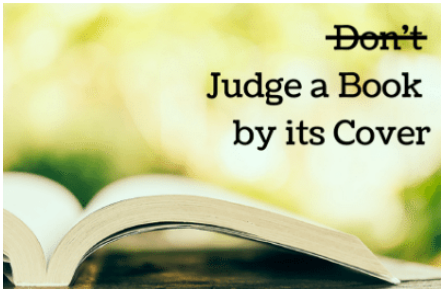

Image Source

For the non-competitive part, train your model using `train_x.csv` and `train_y.csv`, and test on `non_comp_test_x.csv`, and then report your accuracy by comparing the predicted labels with ground truth set of labels given in `non_comp_test_y.csv`. For the competitive part, you can use both the above splits for training the model, and submit predictions on `comp_test_x.csv` (target labels are not provided for this set).

2 Non-Competitive Part (30 Marks)

1. **Convolutional Neural Network (15 marks)** : For this part you have to implement a simple CNN which takes as input book cover images and returns the genre label. Implement a Convolutional Neural Network with the following structure:
   - CONV1: Kernel Size → 5x5, Input Size → 3, Output Size → 32
   - POOL1 : Kernel Size → 2x2
   - CONV2 : Kernel Size → 5x5, Input Size → 32, Output Size → 64
   - POOL2 : Kernel Size → 2x2
   - CONV3 : Kernel Size → 5x5, Input Size → 64, Output Size → 128
   - POOL3 : Kernel Size → 2x2
   - FC1 : Fully Connected Layer with 128 outputs
   - FC2 : Fully Connected Layer with 30 outputs

   Use ReLU as the activation function for all layers apart from the Pooling layers.

   Use the book cover images as input for this CNN and train the model. Finally, report the accuracy you get on the non-competitive test set.
2. **Recurrent Neural Network (15 marks)**: For this part, you have to implement a bidirectional RNN which takes as input book titles and returns the genre label. Tokenize the input text using any tokenizer of your choice and then obtain GloVe Embeddings for each word (specifically, glove_6b_300d). We recommend using torchtext to obtain these embeddings. The

tokenizer converts each sentence into a list of word vectors. Since each sentence may have different number of words, use padding/trimming to make all list sizes consistent. Implement a neural network with the following architecture:

- Embedding Layer : Initialize this with the vocabulary vectors from the pretrained embedding you use.
- RNN layer : Hidden layer size → 128, bidirectional → True, batchfirst → True
- MLP layers
  - FC1 : Fully Connected Layer with output 128
  - FC2 : Fully Connected Layer with input 128 and output 30 (number of classes)

Use the tanh activation function for the MLP layers. Again, report the accuracy on the non-competitive test set.

3 Competitive Part (70 marks)

In this part, you are free to train any model architecture of your choice, which takes as input either/both of cover images and titles. The score for this part will depend on your performance relative to other groups in the class.

You are free to use any generic pre-trained models or embeddings, but you are not allowed to use any task-specific models available on the web. You are not allowed to scrape additional information about the data from the internet. You can use standard open-source libraries like Torchvision, Torchtext or Hugging Face. **For using other libraries or if in doubt, please clarify with the instructor or TAs.**

You are not allowed to download any additional training examples from the internet, you should only work with the given dataset. **Any augmentations/preprocessing you do must be in your training scripts**. Only for the competitive part, you are allowed to train your model on both the training and non-competitive test set. **We may train your model again using your submitted script and any significant deviations will be severely penalised.** (Please make sure to minimize any randomness in your models by initializing seeds etc. so that the models can be replicated if needed)

The kaggle contest is hosted here. To access the competition, please look out for the invitation link on piazza. The final score will depend on ranking in both the public leaderboard (consisting of 30% test set) and private leaderboard (consisting of 70% test set). The private leaderboard will be visible only after the assignment deadline.

4 Submission instructions

The submission will consist of

1. **Kaggle competition**: Register for the competition and form a team. Upload the test labels directly as per the instructions given on the competition page. Please note, you will not receive a score in the competitive part if your submission is not visible on the leaderboard.

2. **Moodle submission**: Only one submission per group is needed. Upload all python files alongwith the writeup in a zip file named <`entrynum1`>_<`entrynum2`>`.zip` or <`entrynum1`>`.zip` depending on the group size. There should be the following files (alongwith the writeup pdf, and any other additional python files as needed):

   a) *requirements.txt*: Make sure to include the pytorch version, torchvision version etc here. Any non-trivial libraries you use should be here to make sure there is no version mismatch while running your code. This file would be run as `pip install -r requirements.txt`.

   b) *cnn.py*: Would be run as `python3 cnn.py` <`dataset_dir_path`>. This should train the CNN and generate a file <`non_comp_test_pred_y.csv`> containing non-competitive test set predictions.

   c) *rnn.py*: Would be run as `python3 rnn.py` <`dataset_dir_path`>. This should train the RNN and generate a file <`non_comp_test_pred_y.csv`> containing non-competitive test set predictions.

   d) *comp.py*: Would be run as `python3 comp.py` <`dataset_dir_path`>. This should train the model used in the competitive part and generate a file <`comp_test_y.csv`> containing competitive test set predictions.

5 Important links

- Kaggle competition: https://www.kaggle.com/competitions/col774-2022 (invitation link to be made available on piazza)
- Dataset: https://www.kaggle.com/competitions/col774-2022/data