CS360 Machine Learning Final Competition
Phase 1 Due: May 2 23:55, 2024
Competition Over: May 10 23:55, 2024
Phase 2 Due: May 13 23:55, 2024

April 23, 2024

General Instructions.

This homework is a Kaggle Competition: you will train neural nets using Pytorch on a given dataset, and submit your results on the Kaggle competition website to get evaluated.

This is an individual assignment/competition, which means each student must hand in their own answers, and you must write and use your own code in the programming parts of the assignment. It is acceptable for you to collaborate in figuring out answers and to help each other solve the problems, and you must list the names of students you discussed this with. We will assume that, as participants in an undergraduate course, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.

Your score will be mostly dependent on your model's performance. You are also required to write a short report (1-2 pages) to document what you have done to improve your model in the competition. We recommend you type (or very neatly hand write) answers to the write-up part — for answers including figure illustrations, you can draw by hand and take photos. You are welcome to use Latex, Microsoft Word, and Google Doc, as long as the submitted format is pdf.

There are **two phases of scoring**. You only need to submit your Kaggle name for the first phase. **You get 10% of the total score of this final competition for the first phase, and you need to achieve at least 50% accuracy on the leaderboard to get full credit**. It is easy to achieve 50% accuracy. We set the first phase to encourage you to start early.

For the second phase, please submit **exactly two files**: (1) a **pdf file** that is the report for your model; and (2) a **zip file** which consists of your code as well as your model (the model refers to the trained model, not the code for the model). The zip file can contain multiple code files. **Please do not submit any fewer or any more than these two files, the pdf and the zip files. Please make sure we can run your code, and document any extra steps required (e.g., installation of certain libraries) to run your code. If you are using ipython notebook, please keep all the generated logs.** All files should be submitted to Brightspace (we don't submit to gradescope this time). **Neatness and clarity count!** If we can't understand your report, or if your report is not consistent with your model/competition performance, you will not receive full credit.

Also note you can NOT submit your predictions to the Kaggle competition after the deadline. The system shuts down right at the deadline, and the best result from all your submissions up to that point is used for evaluation. You have some extra days to finish your report and submit it to brightspace.

1 Final Competition

The Kaggle competition is held at `https://www.kaggle.com/t/e6112623017f4586bc3bd98e99b23087`. You'll need to create a Kaggle account to participate in the competition. **In your report, indicate your name of your Kaggle account so we can map to your model's performance.**

1. *Datasets and Scripts:* Download the datasets from the "Data" tab. There are three data files:

   (a) train_mp3s.tar : a tarball file containing over 11k mp3 song snippets. Every snippet if of a length of 3 seconds.

   (b) train_label.txt : the corresponding labels for the train mp3 files. The labels are of 4 categories: no voices in the snippet, one male-like voice in the snippet, one female-like voice in the snippet, and more than one person's voice in the snippet.

   (c) test_mp3s.tar : a tarball file containing over 3k mp3 song snippets for your model to make predictions. You will need to generate the labels and submit them to the Kaggle website to get scored.

   We provide an example_submission.csv file with the submission file format. Your submission file should contain two columns, with column names (id, category), where "id" is the index of the data point (starting at 0) and "category" is your predicted class (also starting at 0). For example, a row of "3,2" indicates that your model predicts the fourth data point to be of class 2.

   Disclaimer: The song list is generated with ChatGPT, and we do not choose the distribution of male-like vs. female-like voices. The course staff and some CS faculty (Prof. Wen and Prof. Tan) label the song snippets. There could be potential noises in labels due to human error or inconsistencies among different labelers. Also, when we label the snippets, we first remove the background music and only label the voices part. However, in the data we give you, we add back the background music without changing the labels.

2. *Data Preprocessing:* You should design your own data preprocessing method. As a starting point, you may want to use librosa to process the mp3 file. **Please include your data processing code in your code submission, and in the report describe how you preprocess the mp3 data.**

3. *Models:* You are free to use any machine learning model, and we recommend you use deep models. **In your report, give an illustration of your best model structure. For example, use a flow chart to show that the input data first goes into a convolutional layer, followed by ReLU activation, then followed by .... Also, report (an estimate of) the number of hours you spend training your model and which GPU(s) you use.** If you are not using deep models, provide similar information to illustrate your model design and training costs.

4. *Training:* You should randomly select a portion (e.g., 3000 data points) of the training data we give you as the validation set to tune the hyper-parameters. I.e., you train the model on the rest 8k data points, and check the performance on the selected validation set. Then, you change the hyper-parameters and train again on the 8k data points. You repeat this process until you are happy with the validation set performance. At this point, it is optional and, in many cases, can get you some boosted performance that you train on the entire 11k data points in the train_mp3s.tar using the best hyper-parameters you find, and then generate the predictions on the test_mp3s.tar dataset. **In your report, document your training procedure and discuss the hyper-parameters/optimization methods/data augmentation you have tried (if any).** Data augmentation refers to the procedure of applying random distortions to the data without altering the labels. For example, you can add some echo effects to the music and the label should not change. Typically, you can get quite significant improvements by applying data augmentations.

5. *Submission:* As stated above, you can follow the example submission file example_submission.csv to create your own submission, and submit it under the "Submit Predictions" tab on the web page. Note that you are **only allowed to make 5 submissions each day** (to avoid manually over-fitting to the test dataset) and your best submission is used for your final score (you can also manually select the model to be scored).

6. *Evaluation:* The evaluation metric for this competition is the test set accuracy, namely, the percentage of correct predictions divided by the total number of data points. We will release a baseline model performance based on how the entire class is doing. Exceeding the baseline model accuracy will score 50% pts of the **second phase scoring**. The rest 50% pts will be scored based on your model's

performance. Here's the tentative scoring mechanism (we may change it depending on the overall performance of all submissions): The average accuracy (call it x^*) of the top 3 submissions will get 100%. The baseline model accuracy (call it $\bar{x}$) will get 50%. Denote your accuracy as x, and your score will be

$$score(x) = \min\{\max\{50 * 2^{(\frac{x-\bar{x}}{x^*-\bar{x}})}, 0\}, 100\} \tag{1}$$

The general idea is that getting higher accuracies typically requires an exponential increase in effort/computation. Here's an example scoring, say $\bar{x} = 0.5$, $x^* = 0.8$, and $x = 0.75$, your score will be ≈ 89.08.

We will give extra points for exceptional performance.

Please note that the final report you submitted and your code are used to validate your model's performance. If you don't include the information we have asked for or your results cannot be reproduced (we will not check for exact reproduction; a reasonably similar performance is good enough), we will apply some penalties to the final score.

7. *Rules:* This is an individual assignment/competition and everyone has to train/submit a model, and your models should not be duplicates of each other.

   Your model should be a Pytroch model if you choose to use deep models.

   Don't copy a pre-trained model online.

   Don't manually label the test datasets. All predictions have to be made by a machine learning model. **Don't train on test!**

8. *Helpful Materials:*

   (a) Pytorch Convolutional Networks Tutorial on CIFAR-10: `https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html`.

   (b) You can use your own GPUs for this task. You can also use the Google Colab or the Kaggle platform, which offers free GPUs.
   Google Colab tutorial:
   `https://colab.research.google.com/drive/16pBJQePbqkz3QFV54L4NIkOn1kwpuRrj`.
   Kaggle notebook:
   `https://www.kaggle.com/docs/efficient-gpu-usage`.

   (c) Pytorch save and load models: `https://pytorch.org/tutorials/beginner/saving_loading_models.html`

   Pytorch optimizers (also pay attention to the learning rate schedulers): `https://pytorch.org/docs/stable/optim.html`

   Pytorch bathnorm layer: `https://pytorch.org/docs/stable/generated/torch.nn.BatchNorm2d.html`

   Pytorch data augmentation for image data`https://pytorch.org/vision/stable/transforms.html`

Have Fun!