COMP 4900 Introduction to Machine Learning
Homework 3

Winter 2020
School of Computer Science, Carleton University
Professor: Majid Komeili

This mini-project is to be completed in groups of two. You will submit your assignment on CuLearn as a group. You must register your group on CuLearn.

Introduction

This assignment will give you experience with working on image analysis prediction challenge. You will realize a 10-class classification technique to classify a dataset which is based upon Fashion-MNIST.

Here, you will be working with a Modified Fashion-MNIST dataset that we have constructed. In this modified dataset, each image contains three articles, and the goal is to output the class label of the most expensive article presented in the image. Articles, from the least to the most expensive, are: T-shirt/top (0), Trouser (1), Pullover(2), Dress (3), Coat(4), Sandal (5), Shirt (6), Sneaker (7), Bag (8), Ankle boot (9); where the number within parenthesis shows the class label associated to each article. Examples of this task are shown in Fig. 1.

Note that this is a supervised classification task: Every image has an associated label (i.e., the class of the most expensive article) and your goal is to predict this label.

Download the training and test data from the Kaggle link provided on CuLearn. `Train.pkl` includes 60,000 training images where the class label for each image is provided in the `TrainLabels.csv` file. `Test.pkl` includes 10,000 test images. You need to generate a `.csv` file that includes your predicted labels for the test images. Use `TrainLabels.csv` besides `Train.pkl` to train your models.

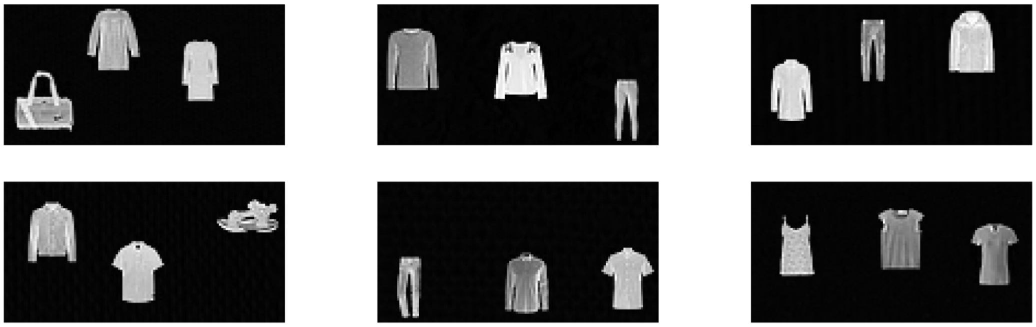

Figure 1: Example images from the dataset. For example, the target label for the top-left image would be 8, while the target label for the bottom-right image would be 3.

Your tasks

You must design and validate a supervised classification model to perform the Modified Fashion-MNIST prediction task. There are no restrictions on your model, except that it should be written in Python. As with the previous mini-projects, you must write a report about your approach, so you should develop a coherent validation pipeline and ideally provide justification/motivation for your design decisions. You are free to develop a single model or to use an ensemble; there are no hard restrictions.

You must run your model on the test data provided in `Test.pkl` and submit the result on Kaggle competition. (See below for more details)

Kaggle competition

`Test.pkl` contains test images. You need to make a prediction for all test images and submit a `.csv` file in Kaggle where each line contains the corresponding image index (between 0 and 9999) and predicted label which is an integer number between 0 and 9.
`ExampleSubmissionRandom` is an example of the kind of `csv` file you should submit in Kaggle. (Note: in the example file, the class label written in each line ,in the second column, is just a random value and you should replace them with your predicted values). You are limited to two submissions per day. You must register in the Kaggle competition using your `carleton.ca`. If you already have a Kaggle account under a different email address, do not delete your account. You only need to change your email in your Kaggle profile. **You must form a team in the competition page on Kaggle, the name of your team must be the same as the name of your group on CuLearn (e.g. Team Name: Group 10).** Except where explicitly noted, you are free to use any Python library for this project. You are not allowed to use any training data other than what is provided for the competition.

Report

We are flexible on how you report your results, but you must adhere to the following structure:

Abstract (100-250 words): provide a summary of the project task and highlight your most important findings.

Introduction (at least one paragraph): Summarize the project task, the dataset and important findings. This is similar to the abstract but you should provide more details.

Dataset (at least one paragraph): Briefly describe the dataset. Also, describe the preprocessing steps (if you have any) for preparing the feature vectors.

Proposed approach: Briefly describe the methods you have implemented or used for this project. No need to provide detailed derivations and proofs but you need to provide some background, description and motivation for each model. You should properly cite and acknowledge previous works/publications that you use or build upon. Discuss any decision about training/validation splits, algorithm selection, regularization, hyper-parameters, etc.

Results: Summarize your results using tables and/or figures. Discuss the results for each model (for example accuracy, runtime, etc.). Since you do not have the labels for the test set, your results should be based on your validation set(s). Report your test set leaderboard accuracy too.

Discussion and conclusion: Discuss and summarize the key takeaways from the project and possible directions for future investigation.

Statement of contributions: (1-3 sentences) State the breakdown of the workload across the team members.

You are expected to discuss your findings with scientific rigour.

The report is limited to 5 pages (single-spaced, minimum font size of 11 and 1 inch minimum margin each side). Imagine you are writing a paper for a conference. We highly recommend to use LaTeX for preparing your report.

Appendix To facilitate the grading process, attach the codes for your implementation to the end of your report. This does **not** count towards the page limit of the report.

Deliverables

- `report.pdf`: Your report as a single pdf file.
- `code.zip`: Your codes (e.g. .py, .ipynb, etc.) which must work with Python 3.6 in Colab. Include a `readme` file and provide instruction for TA on how to replicate your results on Colab. All the results including your leaderboard submission must be reproducible in Colab using the submitted `code.zip`. Points will be deducted if we have a hard time reading or understanding the structure of your code. **Do not include report.pdf in the code.zip file.**

Evaluation

This is an open-ended project. The evaluation has two parts each worth 50 points out of 100.

Performance (50 points): This is based on the performance of your best model on the held-out test set on the Kaggle competition. Your grade will be computed based on a linear interpolation between three points: the 2nd top group, a TA baseline and a random baseline. The random baseline is the score needed to get more than 0% on the competition. The TA baseline is the score needed to get 75% on the competition. In other words, if your score is between the random and TA baseline, your grade is a linear interpolation between 0% and 75% on the competition; likewise, if your score is between the TA baseline and the 2nd best group, your grade will be between 75% and 100% on the competition. In addition to the above criteria, the top two groups all receive 100%. Additionally, the top group will receive 10% points as bonus.

Quality of your report and proposed methodology (50 points): Your report should be both thorough and concise. It will be judged based on its scientific quality including but not limited to: Does the report include all the required experiments? Is the report technically sound? How thorough/rigorous are your experiments? Is the report well-organized and coherent? Is the report clear and free of grammatical errors and typos? Does the report contain sufficient and appropriate references and related work?

All members of a group will receive the same mark.

Final remarks

You are expected to display initiative, creativity, scientific rigour and critical thinking skills.

You can discuss methods and technical issues with members of other teams, but you cannot share any code or data with other teams. Any team found to cheat (e.g. use external information, use resources without proper references) on either the code, predictions or written report will receive a score of 0 for all components of the project.