Deep Learning - Project 2021

Value: 60% - Due Date Friday 1st May 20:00

Version 1.0

Overview

This is the specification for the 2021 delivery of the Deep Learning class.

The goal of the project is to perform a systematic investigation of a number of Deep Learning methods in the context of text processing tasks, and benchmark these methods against classical methods where appropriate.

This project has been designed to give you a range of task elements that you can use to build up your skills in Deep Learning. As such this project specification is detailed and descriptive.

The outputs for this project will be a detailed report (between 11 and 15 pages formatted as per this document), source code notebooks, and trained models. Detail on what is to be included in each of these pieces is detailed below. The report is to be submitted via Brightspace with a code archive (.zip or .tar.gz),. The code archive should not include your trained models - instead it should where needed download your models from a publicly accessible link within your code.

Your implementation should be made in Keras / TensorFlow with Notebook files such that your code can be opened and run within Google Collab -- though it is noted that training may in practice need to be run on a local machine for full training to take place.

Note that you cannot use any alternative data set.

Task Specification

Part 1: Lyric Genre Prediction Task - 40% (of project total)

The core of this project is based around a simple task -- performing genre analysis on the “Multi-Lingual Lyrics for Genre Classification” dataset on kaggle.

https://www.kaggle.com/mateibejan/multilingual-lyrics-for-genre-classification?select=train.csv

This is an extensive dataset that is split into training and testing subsets. The testing dataset should be used for final testing only. The training dataset should be all training and validation tasks as appropriate.

This first part of the task is to perform a number of analyses based on training from scratch and the use of pre-trained models to predict genre based on song lyrics only.

Below I've set out a number of different comparisons that you are to perform. You can think of these as being individual chunks of analysis. These do not necessarily build on each other. It is up to you to decide whether these are done in isolation or not. The key point is that you understand yourself what you are doing, and that this understanding is reflected both in well documented code, and in your report.

Your models should be designed to minimize overfitting as appropriate.

In all cases you should record your results as graphs for Training and Validation data and report the test result after training has been completed. Please select whatever loss functions or metrics you think are appropriate based on notes provided in class, or more widely based on what is appropriate in a text classification task.

RNN Variants
The first sub-assignment is to compare performance on the classification task across Recurrent Network Variants. Specifically compare LSTM and Basic RNN models. You are free to choose your own state size for the recurrent network, however please use the same state size for both RNN variants.

Also compare a single layer LSTM implementation to a multi-layer LSTM implementation.

Embeddings
Distributed embeddings provide a lot of power in text classification, but there are many different Embeddings types that can be used. Compare classification between Embeddings learned on the fly to any pre-trained word embedding available from the Tensorflow Hub.

CNN for Text Classification
As mentioned in the lecture notes, CNNs are designed to model local features while LSTMs are very good at handling long range dependencies. Investigate the use of CNNs with multiple and heterogeneous kernel sizes both as an alternative to an LSTM solution, and as an additional layer before a LSTM solution.

Model Saving
From the various models above, save two models. One should be the best model you obtained without pre-trained embeddings, and the other should be the best model you obtained with pre-trained models.

A link to these best performing models should be included in your submission and a demonstration notebook (described later). You will also be using these saved models in Parts 2 and 3 below for training. There are many ways in which models can be saved. I'm not prescribing a specific way this is to be done. You are free to use whichever method you find most suitable. As always clearly document your design.

Part 2: Transfer Learning - 35%

While we can use one task to train a model, we can sometimes make use of elements of this model to bootstrap the training of a different model. This is referred to in general as Transfer Learning and may or may not involve fine tuning of the model for the new task or domain.

Given this idea, your goal in this part of the assignment is to re-use the models saved from the last part of the assignment and apply them to a new task. To make your life a little easier the task that we will apply the new model to is also based on the same lyrics dataset -- but it is important to keep in mind that in general Transfer Learning can be applicable to entirely new datasets.

Your goal is to create a new task based on predicting the artist based on lyrics. For this task you should rework the datasets (training and testing) to build a dataset consisting of the top 10 most frequent artists and then split that whole dataset into training and validation datasets (there will be no test dataset for this particular task). 10% of the data should be used for validation.

For the newly reworked artist dataset, build and evaluate models based on your best 2 models obtained from part 1 using Transfer Learning method and also build models from scratch based on the new artist dataset only. The models you build should allow some amount of like to like comparison between the newly created models and the models imported from Part A.

Save the best performing resulting Transfer Learning and ‘From Scratch’ models for this dataset. Links to these models need to be supplied as part of your submission. Your evaluation of the models should be based minimally on training and validation error and any other metrics or methods you think appropriate.

Part 3: Writing your own Lyrics - 25%

Practical language processing tasks aren’t just about classifying. In the second part of the assignment you will put your skills in RNNs and related technologies to work to generate some original lyrics and benchmark your model against a more classical implementation.

For this work make use of the lyrics dataset but let’s split the data differently. For each of the examples below, build one model for each of the top three most frequent genres. Keep in mind that we do not need to use a training validation and testing split of the data in this case.

Your core model should be based on the use of LSTMs, but beyond this you are free to explore whatever architecture and hyper-parameter variants that you find results in the best performance in the language generation task.

With the same data, also implement a statistical model for language generation.

Report model performance in terms of perplexity and any other metrics or methods you find appropriate. Provide 5 outputs each from your best implementation and the statistical model. Make sure to save your best model and provide it a link in the submission.

Document Specification

Your document should be 11-15 pages in length and formatted as per the current document, i.e., standard margins, 11pt Arial font, 1.15 spacing between lines, and a 50% spacing (approximately) between paragraphs.

The first page is a standalone cover page. The cover page should include:

- Your name
- Your student number
- A link to a zip file that contains all your code (not models) collected data files etc.
- The following statement:

“I confirm that the document and related code here is all my own work and that I did not engage in unfair practice in any way.”

Your code should be submitted as Jupyter notebooks. There should be two notebooks only. Training.ipynb should contain all code for training your models and Testing.ipynb should demonstrate the application of all models that you have trained, i.e., it should be capable of loading the models that you were requested to save and demonstrate them being used to classify on the datasets or generate text as appropriate. (You can if necessary use separate training notebook files for each of the three individual assignment parts).

This report should be a well written, suitably structured and appropriately referenced document that justifies decisions, provides results as well as your analysis. The report should not give a background explanation of Deep Neural Networks or related models.

The following sections should minimally be included:

- Lyrics Modelling Task
- Transfer Learning
- Writing my own Lyrics
- Discussion

Appendices should be avoided.

Submission Details

This assignment is due for submission at 8pm on Friday 1st May. Extensions will not be granted except in the case of documented and approved personal circumstances. Late submissions will

be deducted 2 percent per day for each of the first two days late, and then 1 percent per day for each day thereafter until a final deadline of May 15th.

All taught (not research) students should be available to answer questions on their assignment on Thursday 14th and Friday 15th of May. Arrangements for this will be made later, but are expected to be on MS Teams.

Marking Rubric

Marks for Parts 1, 2 and 3 of this assignment are split in the ratio 40%, 35%, 35%. For each Part, marks will in turn be broken down in a split between documentation and quality of modelling / analysis. The Rubric below details how marks will be split out of 100 for each of the three parts.

0 - 19: The student failed to provide a working implementation of code for achieving this part of the assignment -- or an unacceptable amount of code or documentation indicated unfair practice.

20 - 39: The student provided a working implementation of this Part of the assignment, but the documentation or interview failed to show the student had a clear understanding of the methods employed.

40 - 59: The student provided a working implementation of this Part of the assignment, and the documentation and interview showed that the student had a clear understanding of the methods employed.

60 - 79: The student provided a high quality implementation of this part of the assignment with a high quality report that was detailed in analysis and well motivated.

80 - 100: The student provided an excellent implementation of this Part of the assignment with a report of quality that is worthy for submission for publication at a national conference. To reach this grade the work will generally need a non-trivial original element that is not already covered by the assignment specification -- this is in addition to meeting the other criteria.

All students should be able to defend / explain their submission orally if requested to do so. Any failure to do so will be judged as a failure in the documentation to show the student had a clear understanding of the methods employed. It should be noted that the interview does impact on the marking scheme.

Unfair practice of any type is not accepted. Do not copy text into your report without appropriate citations. Even with citations, no more than 10% of a report should be based around existing material. Note that this also includes ‘self plagiarism’. Similarly, no copying blocks of code into your code without appropriate acknowledgement. Again, even with acknowledgements, only a small portion of code should be copied directly. Unfair practice of this type will be subject to a grade of 0 on the assignment. If you are in doubt - ask me.

For reference here is the Student Union’s guide to unfair practice

https://www.tudublinsu.ie/advice/exams/breachesofregulations/

Group Submission

Group submission of this project is possible. A group submission is a single submission made by one person on behalf of a team of between 2 and 4 people from the same class cohort (i.e., PhD or MSc). It is not possible to have a team with a mix of PhD and MSc students.

There is a 2% discount on the total grade for each person that groups together. For example a team of 2 has a 4% discount, a team of 3 has a 6% discount, and a team of 4 has a 8% discount. Team members should contribute equally to all major aspects of the project.

Each team member does however need to be able to attend an interview on the submission and answer all questions on that submission. In other words, while a group can work on a submission, it is not possible for a person to avoid understanding how any part of the submission works.

In the case of a group submission two additional criteria hold:

- Each team member needs to submit a 5 minute presentation (recorded with voice over) to give an overview of their own learning experience from the project highlighting what they believe worked well in the analysis and what did not.
- A peer review survey between group members also needs to be completed. The peer review survey results in a weighting that can adjust individuals within a group up and down.