Assignment 5, Applied ML

John MacLaren Walsh

Fall, AY 2020-2021

1 Your Challenge

Your task is to build and train a model that predicts whether or not a review for a movie indicates the movie was liked, or was not liked.

2 Training Dataset

Train your model using the IMDB dataset.

3 What you must submit

You will upload to bblearn a file assignment5.zip or assignment5.tgz containing the following files

1. buildAndTrainIMDBModel.py: This must contain all of the code necessary to train your model and then save it.
2. Your saved model.
3. runModel.py: This must contain all of the code to load your saved model, and include a function which takes in a collection of IMDB reviews and scores each of them with your model, returning the models predicted probability that this is a positive review and the models predicted probability that this is a negative review.

4 How you will be graded

If your code works and it appears that your training code generated your model, the rest of your grade will be determined by a relative rank of the loss obtained by your model on a selected testing set.

5 Bonus Assignment

Complete this assignment and get up to 25 extra points on your worst assignment's submission.

5.1 Your Challenge

Your task is to build and train a model capable of predicting the title of a wikipedia article from its text.

5.2 Training Dataset

To train your model, you may sample the English wikipedia dataset from the tensorflow_datasets python package.

5.3 What you must submit

You will upload to bblearn a file assignment6.zip or assignment6.tgz containing the following files

1. buildAndTrainWikipediaModel.py: This must contain all of the code necessary to train your model and then save it. You are free to use transfer learning for this assignment by downloading a pretrained model from tfhub.

2. preprocessDefinition.py: Unlike the previous assignment, the function preprocess, that this defines will be a function you apply to a tensorflow dataset that returns another tensorflow dataset. The incoming dataset will be identically formatted to the English wikipedia dataset from tensorflow_datasets, but will contain fewer elements and some new instances. The outgoing dataset should be compatible with your model. The outgoing titles (labels) should also be in an identical tokenized format consistent with your model. The function should tokenize words, and can truncate articles. It should also truncate/pad titles to the same length of 10 words.

3. Your models saved in .h5 format.

4. topTwentySentences.py: Loads your model into a function that can be applied via .map on the dataset produced by preprocessDefinition.py to produce your estimate of the 20 most likely titles for the article text, in order of decreasing ranked probability under your model.

5.4 How you will be graded

We will build a combined score based on the top 5, 10, and 20 error rates for your predicted titles. Not all of the data used to evaluate your model will be within the english wikipedia dataset in tensorflow_datasets, so be sure to protect yourself against overfitting by using your own separate training, validation, and testing subsets.