CPSC 585 - Artificial Neural Networks

Project 3 - Spring 2024

In this project, you will work with a team to do [sentiment analysis](https://en.wikipedia.org/wiki/Sentiment_analysis) on comments left by students at Rate My Professors.

# Project Teams

This project must be completed in a team of four or five. The instructor will assign teams for this project in Canvas.

See the following sections of the Canvas documentation for instructions on group submission:

* [How do I submit an assignment on behalf of a group?](https://community.canvaslms.com/t5/Student-Guide/How-do-I-submit-an-assignment-on-behalf-of-a-group/ta-p/294)

# Datasets

## Embeddings

You are free to experiment with your own embeddings, but you will probably want to use a pre-trained embedding like [GloVe](https://nlp.stanford.edu/projects/glove/) or [ELMo](https://allenai.org/allennlp/software/elmo).

To use GloVe from Keras you will need to download the dataset and use it to create an Embedding layer. See [Using pre-trained word embeddings](https://keras.io/examples/nlp/pretrained_word_embeddings) in the Keras examples. PyTorch, on the other hand, includes GloVe in the torchtext.vocab module. See [Pretrained Word Embeddings](https://pytorch.org/text/stable/vocab.html#pretrained-word-embeddings).

*Note*: Do not use wget and unzip to download datasets each time as shown in the example. Upload a copy to [Google Drive](https://colab.research.google.com/notebooks/io.ipynb#scrollTo=c2W5A2px3doP), or [save and reload the model](https://www.tensorflow.org/guide/keras/save_and_serialize) once the embedding layer has been created.

Alternatively, you might use a third-party library like [Gensim](https://radimrehurek.com/gensim/).

## Comments and ratings

In addition to leaving comments on Rate My Professors, reviews also include scores for “Quality” and “Difficulty.” There are at least two publicly-available datasets that include this information:

* A [larger dataset](https://data.mendeley.com/datasets/fvtfjyvw7d/2) in [CSV](https://docs.python.org/3/library/csv.html) format from [Dr. Hibo Je](https://data.mendeley.com/datasets/fvtfjyvw7d/2) at Tsinghua University.
* A [smaller dataset](https://www.kaggle.com/datasets/tilorc/rate-my-professor-reviews-5c-colleges) in [JSON](https://docs.python.org/3/library/json.html) format from Kaggle containing reviews from the undergraduate Claremont Colleges.

You may also wish to use additional data for training, whether through [data](https://neptune.ai/blog/data-augmentation-nlp) [augmentation](https://blog.paperspace.com/data-augmentation-for-nlp/) or experimenting with [web](https://data.mendeley.com/datasets/vvcj4f7smy) [scraping](https://pypi.org/project/RateMyProfessorAPI/) [code](https://youtu.be/mWUOdV2nMOk).

# Tasks

## Create embeddings for the dataset

While words can be represented using [one-hot](https://en.wikipedia.org/wiki/One-hot) encoding, your analysis is likely to be significantly more effective with a [word embedding](https://en.wikipedia.org/wiki/Word_embedding). Begin by using a pre-trained embedding model, but once you have complete the tasks below, you might wish to experiment with fine-tuning the embedding layer for the given task.

## Construct a Recurrent Neural Network

Construct a recurrent neural network model to predict the quality and difficulty scores that a student will assign, given the text of the student’s comments.

You are free to use any appropriate RNN architecture, and are welcome to attempt transfer learning with a pre-trained model upstream. To get started, see the Keras tutorial [Text classification with an RNN](https://www.tensorflow.org/text/tutorials/text_classification_rnn) or the PyTorch tutorial [Text classification with the torchtext library](https://pytorch.org/tutorials/beginner/text_sentiment_ngrams_tutorial.html).

*Notes*:

* While the examples are for text classification, predicting numerical ratings might be better defined as a regression task. Be sure to consider both approaches in your network design and model evaluation.
* While the task is clearly defined, this project is more open-ended than previous projects.. Be sure to document all methods and results that your team explores, even those that don’t work out.

Experiment with varying the number of units to see how the effectiveness of the prediction varies with input sequence length.

## Experiment with Transformer encoders

Construct a network using Transformer encoder blocks instead of RNN layers for the same prediction task. Remember that this network will require position encoding for sequences as well as word embedding.

If you are using Keras, the [Text classification with Transformer](https://keras.io/examples/nlp/text_classification_with_transformer/) shows how to construct an encoder block from position embedding and multi-head attention layers, but you can also use the [PositionEmbedding](https://keras.io/api/keras_nlp/modeling_layers/position_embedding/) and [TransformerEncoder](https://keras.io/api/keras_nlp/modeling_layers/transformer_encoder/) layers directly. For PyTorch, the tutorial [Language Modeling with nn.Transformer and torchtext](https://pytorch.org/tutorials/beginner/transformer_tutorial.html) shows how to construct both encoder and decoder blocks.

Compare the performance of your Transformer with the performance of the RNN.

# Platform

Perform the tasks above and document their results using a notebook on [Google Colab](https://colab.research.google.com/) with your @csu.fullerton.edu account. If you are not familiar with Google Colab or Jupyter Notebooks, the [Welcome To Colaboratory](https://colab.research.google.com/) notebook should help you get started. Note, in particular, the section [Using Accelerated Hardware](https://colab.research.google.com/#using-accelerated-hardware).

While you may choose to work locally, especially if you have access to a physical machine with a GPU, your project submission must be uploaded to Google Drive and run successfully in Colab.

## Libraries

In addition to [TensorBoard](https://www.tensorflow.org/tensorboard) you are welcome to use other libraries such as [Gensim](https://radimrehurek.com/gensim/), [spaCy](https://spacy.io/), [fast.ai](https://docs.fast.ai/) or [PyTorch Lightning](https://pytorch-lightning.readthedocs.io/) and collections of pre-trained models such [Model Zoo](https://modelzoo.co/) or [Hugging Face Hub](https://huggingface.co/docs/hub/index) if you find them helpful.

# Documenting your results

Notebooks allow you to create documents mixing text, equations, code, and visualizations. Your project should make good use of these features. For example:

* Identify each task to be performed, documenting any decisions made.
* Include both the code to perform each task and its output. Where appropriate, tasks should be broken up into separate blocks, with the results shown for each.
* Include written analysis of results along with code output and visualizations.

In short, a reader unfamiliar with the project should be able to read your notebook and understand what you did and what results you obtained.

For help with rendering equations in your notebook, you may wish to consult this [LaTeX Equation Cheat Sheet](https://colab.research.google.com/drive/1VAYafeBSGo0saKr3BcxMJkXHfSEQFo-Z?usp=sharing).

At the end of the notebook, assign a percentage contribution to each member of the team. If your team cannot reach an agreement on each member’s level of contribution, describe the tasks that each member performed.

# Submission

From inside your Google Colab notebook, use the **Share** button at the top-right of the toolbar to [share your notebook](https://colab.research.google.com/notebooks/basic_features_overview.ipynb#scrollTo=aro-UJgUQSH1) with the professor:

1. Make certain that you are logged into Colaboratory with your @csu.fullerton.edu email address.
2. Add the professor’s @fullerton.edu email address as an **Editor**.
3. Leave **General access** set to ***Restricted***. Do not set it to ***Cal State Fullerton*** or ***Anyone with the link***.
4. Use the **Copy Link** button to copy the link to the clipboard.
5. [Submit the link](https://community.canvaslms.com/t5/Student-Guide/How-do-I-enter-a-URL-as-an-assignment-submission/ta-p/286) you copied via Canvas by the deadline.

## Grading

The project itself will be evaluated on the following scale, inspired by the work of Professors Michael Ekstrand at Boise State University and Christopher Ryu at California State University, Fullerton:

**Exemplary (5 points)**

The project is a success. All requirements met. The quality of the work is high.

**Basically Correct (4 points)**

The project is an overall success, but some requirements are not met completely, or the quality of the work is inconsistent.

**Solid Start (3 points)**

The project is mostly finished, but some requirements are missing, or the quality of the work does not yet meet professional standards.

**Serious Issues (2 points)**

The project has fundamental issues in its implementation or quality.

**Did Something (1 point)**

The project was started but has not been completed enough to assess its quality fairly or is on the wrong track.

**Did Nothing (0 points)**

The project was not submitted, contained work belonging to someone else, or was of such low quality that there is nothing to assess.

Individual grades for the project may be adjusted based on levels of contribution.