Assignment #4
COMS 4995.06. Fall 2018.
Due Nov 26th.

About: In this assignment you'll gain experience with TensorFlow.js by training a model locally in Keras and deploying it in the browser. I think this is a valuable skill that will serve you well in the future.

Part 1: Train a text classifier and deploy it in the browser (75 points).

Modify the starter code (**8_colab_to_webpage.ipynb** on CourseWorks) to classify sentences from three books on Project Gutenberg. See the lecture 8 slides for a walkthrough of using this code.

1. First, see if you can configure the starter code to train a model and commit it to GitHub Pages. Visit your website and verify the model works. You should not need to modify the JavaScript or HTML in the starter code.

2. Next, modify the starter code to build a training set from three books (choose them from Project Gutenberg). Extract the first 1,000 sentences or so from each, and use these to train your model.

3. As written, the starter code uses a simple model. Modify this code to improve its accuracy by using a LSTM.

The goal here isn't to train the world's most accurate model, it's just to get the mechanics working and build a working end-to-end demo. You do not need to train a model on every sentence in each book. Instead, use the first 1,000 sentences, or another reasonable small number.

Your submission should include a link to your GitHub pages site running your model, as well as your notebook.

Part 2: Predict color R,G,B values with ColorBot (12.5 points)

Modify the starter code (**8_colorbot_predict_starter.ipynb**) to predict color R,G,B values. You should only need to write several lines of code. Include your completed notebook and the output in your submission. Slide 74 has code that will probably help.

Part 3: Generate color names with ColorBot (12.5 points)

Modify the starter code **(8_colorbot_generate_starter.ipynb**) to generate color names. You should only need to write several lines of code. Include your completed notebook and the output in your submission.

Extra credit

As always, extra credit problems are optional. Feel free to submit solutions to these problems anytime before project presentations on 12/14.

EC 1: Deploy one of your ColorBot models in the browser

- Include a link to your URL with a live demo in your submission.

EC 2: Generate QuickDraw drawings with an RNN.

- Train an RNN to generate QuickDraw style drawings. Using a small subset of the dataset is cool, the goal here is to build a working proof of concept. The loader from assignment 3 should be helpful.