Homework 8.

Due on Gradescope by 5 PM, Saturday, June 5

Submission format. Please submit Google Colab your notebook in **.ipynb** format on Gradescope. Students are recommended to organize their code into modular cells with proper comments for ease of readability.

1. **Training RNNs with different regularization techniques:** Consider the simple RNN tutorial provided by PyTorch: https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html. Follow this tutorial to first code up a simple RNN for generating names, given a language and starting alphabet as input. The estimated time to train a single model is 6 minutes, and you may need to train 5 or more models. You will need to begin by entering the following commands in your colab notebook to download the data:

   ```
   !wget https://download.pytorch.org/tutorial/data.zip
   !unzip data.zip -d
   ```

   The training data that you download from the above link would contain names in 18 languages. Before starting the training:

   - Split the provided data into train and validation set: choose the last 15% of names from each category as validation set (e.g., you can do it by creating a new dictionary *category_lines_val* similarly as *category_lines*, but with last 15% of entries from *lines* variable. You will be using this set to observe differences across various experiments by plotting training curves (loss vs. iterations). Note: you will need to randomly shuffle the *lines* variable in "Preparing the Data" section before assigning a subset of *lines* to validation set.
   - Add a function to evaluate the performance of trained models on the entire validation set and output the average negative log-likelihood (this is the loss being used in the PyTorch tutorial linked above).
   - Modify the main training loop to run your validation function every 5000 iterations.
   - At the very begining of your code, set the random seed to 4 manually using torch.manual_seed(4), numpy.random.seed(4), and random.seed(4).

   Your tasks for this problem are to train the given RNN with different regularization techniques and report the training behavior observed. Specifically:

   (a) The original PyTorch code uses dropout. Begin by training your network with this default setting and show the plot for average validation loss vs. number of iterations.

(b) Next, repeat step (a) but *without* dropout.

(c) Repeat step (a) but this time with L_2 regularization (also known as weight decay) instead of dropout. Note that you will need to search for the optimal weight for the L_2 regularization term. Do the plot as in step (a) for this experiment, after finding the most optimal weight for $L2$ regularization.

(d) Lastly, for all three experiments, generate names for the trained models by providing first letter of **Your First Name** for the following language categories: English, Portuguese, Russian, Greek, Chinese, Dutch.