HW 6 Due: Thursday, May 11, 2023 at 2:00 pm.

Please hand in this Homework as follows:
- Upload to Gradescope `HW6 programming`: A `ipynb` file for problems 1-2.

1. *RNN for predicting characters* (50 points)

   This online post is a good presentation of RNN's and LSTM's

   In this problem you will be experimenting with different RNN architectures to predict characters. The data set is `NLP/tinyshakespeare.txt`. The code is here. In this code there is a generic network that takes as input a 'cell' (an RNN, an LSTM etc.) There is a generic training function, a generic testing function that produces a test loss, and a generic simulation function. Each of these functions gets input a 'cell' that is defined in some function.

   (a) Run the basic LSTM architecture given in the notebook. Plot the error on training as a function of epoch. Try a few priming sentences and see what the trained architecture produces. Load the test set and estimate the error on prediction in the test set.

   (b) Run the same experiment with the basic RNN architecture. This has 1/4 of the parameters of the LSTM. Compare the training error plot with the original LSTM. Compare the test set error.

   (c) Try 3 alternatives. For example an RNN with the same number of parameters as the LSTM. Or an RNN with more than one internal layer. Explain each of your models, plot the training error rate functions as a function of epoch for all your models and show the error rates on the test set as well.

   (d) For each model, run a simulation starting with the same initialization text of your choice.

2. *Classifying spoken digits* ([points)50]

   The data in `Speech/SOUND_DATA_A` in the data folder has the spectrograms for samples of the ten spoken digits. (FYI, Here is a good presentation of how these spectrograms are computed.) All the data are 10 × 64. Ten time steps in the spectrogram and 64 frequencies. One possibility for classification is to run an RNN on the 10 time-steps and use the value of the hidden layer at the final step as input to the classifier. This is a 'mini' version of the supervised RNN we discussed in class.

   (a) Experiment with some of the RNN cells you used for the previous problem to process the speech data. The loss now is different. It is not predicting the data in the next time-step but just predicting the label of the utterance based on the value of the hidden layer at the last time-step. Report the results.

(b) As an alternative, try simply training some of the standard linear classifiers in scikit-learn on the input 640 dimensional data. How do the results compare to the RNN results?

(c) What is your conclusion from the comparison between the two approaches to classification.