CSCI 1051 Homework 2

January 17, 2023

Submission Instructions

Please upload your solutions by **5pm Friday January 20, 2023.** Remember you have 24 hours no-questions-asked *combined* lateness across all assignments.

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be typeset in LaTeX or markdown. I strongly recommend uploading the source LaTeX (found here) to Overleaf for editing.
- Your solutions to coding questions must be written in a Jupyter notebook. I strongly suggest working with colab as we do in the demos.
- You should submit your solutions as a **single PDF** via the assignment on Canvas.

Problem 1 (from January 17)

Part 1

Consider the small recurrent neural network drawn in Figure 1. Suppose we apply the network to a sequence of **even** length T. What values could the final output y_T take? And when does it take each value?

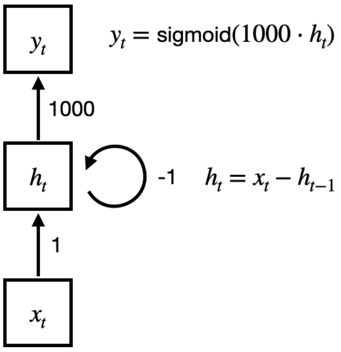

Figure 1: A small recurrent neural network.

Hint: It may help to enroll the network through time on a test input of your choice.

Part 2

Consider the demo on recurrent neural networks. We used a fancy model called an LSTM that we didn't actually implement. In this problem, your job is to replace the LSTM architecture with the simple recurrent neural network from class. In particular, put in the following architecture

$$\begin{aligned}\mathbf{h}^t &= \text{sigmoid}(\mathbf{U}\mathbf{x}^t + \mathbf{W}\mathbf{h}^{t-1}) \\ \hat{\mathbf{y}}^t &= \text{softmax}(\mathbf{V}\mathbf{h}^t).\end{aligned}$$

You may not use the Pytorch recurrent neural network function. Instead, you should be using the following functions: `nn.Linear`, `F.sigmoid`, and `F.softmax`. In addition, `torch.zeros` and `torch.stack` may be helpful.

Hint: The only code block you should change in the demo is the `RNN` class.

Once you have implemented your own recurrent neural network and run the rest of the code, comment on why you think the performance of your RNN differs from the performance of the LSTM.