CSE 527A: Assignment 1

Due: September 27 (Tuesday), 2022

Notes:

- Please submit your homework via Gradescope.
- You can either submit a legibly handwritten or LaTeX generated pdf.
- Make sure you **specify the pages for each problem correctly**. You **will not get points** for problems that are not correctly connected to the corresponding pages.
- Homework is due **by 11:59 PM on the due date**.
- Please keep in mind the collaboration policy as specified in the **Academic Integrity** section of the course syllabus.
- There are 4 problems in the written portion of this assignment and 2 for the coding half.

Problems:

1. (Eisenstein Ch. 3) Design a feedforward network to compute the XOR function:

$$f(x_1, x_2) = \begin{cases} -1, & x_1 = 1,\ x_2 = 1 \\ 1, & x_1 = 1,\ x_2 = 0 \\ 1, & x_1 = 0,\ x_2 = 1 \\ -1, & x_1 = 0,\ x_2 = 0 \end{cases}$$

   Your network should have a single output node which uses the Sign activation function, $f(x) = \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases}$. Use a single hidden layer, with ReLU activation functions. Describe all weights and offsets.

2. (Eisenstein Ch. 3) Consider the same network in problem 1 (with ReLU activations for the hidden layer), $\ell(y^{(i)}, \bar{y})$ being an arbitrary differentiable loss function where $\tilde{y}$ is the activation of the output node. Suppose all weights and offsets are initialized to zero. Show that gradient descent will not learn the desired function from this initialization.

3. (Eisenstein Ch. 6) Prove that n-gram language models give valid probabilities if the n-gram probabilities are valid. Specifically, assume that,

$$\sum_{w_m}^{\nu} p(w_m | w_{m-1}, w_{m-2}, \ldots, w_{m-n+1}) = 1$$

   for all contexts $(w_{m-1}, w_{m-2}, \ldots, w_{m-n+1})$. Prove that $\sum_{w} p_n(w) = 1$ for all $w \,\epsilon\, \nu^*$ where p_n is the probability of w under an n-gram language model. Your proof should proceed by induction. You should handle the start-of-string case $p(w_1 \mid \underbrace{\square, \ldots, \square}_{n-1})$, but you do not need to handle the end-of-string token. (to compute the probability of an entire sentence, it is convenient to pad the beginning and end with special symbols $\square$ and $\blacksquare$).

4. (Eisenstein Ch. 6) First, show that RNN language models are valid using a similar proof technique to the one in problem 3. Next, let $p_r(w)$ indicate the probability of w under RNN r. An ensemble of RNN language models computes the probability,

$$p(w) = \frac{1}{R} \sum_{r=1}^{R} p_r(w)$$

   Does an ensemble of RNN language models compute a valid probability? Please attach the proof.