COS 484: Natural Language Processing **(Due: 02/03/25)**

Assignment #0

Instructor: Danqi Chen, Tri Dao, Vikram Ramaswamy 100 points

Course Policy: Read all the instructions below carefully before you start working on the assignment and before you make a submission. The course assignment policy is available at `http://nlp.cs.princeton.edu/cos484`. When you're ready to submit, please follow the instructions found here: `http://bit.ly/COS_NLP_Submission`

- This assignment contains 2 parts, a theoretical and a programming part. For this introductory assignment, we just have one problem each.
- We *highly* recommended that you typeset your submissions in LaTeX. Use the template provided on the website for your answers. If you have never used LaTeX, you can refer to the short guide here: `http://bit.ly/WorkingWithLaTeX`. Include your name and NetIDs with your submission. If you wish to submit hand-written answers, you can scan and upload the pdf.
- Assignments must be uploaded to Gradescope **before class (1:59pm Eastern)** on the due date mentioned above.
- As per the late-day policy outlined on the course website, you have 4 days to use at your discretion throughout the semester. Once you run out of late hours, late submissions will incur a penalty of 10% for each day, up to a maximum of 3 days beyond which submissions will not be accepted.
- All programming problems in this class will be completed in Google Colab using Python. If you've never worked with Google Colab before, take a look through this introduction guide: `http://bit.ly/WorkingWithColab`. The answers to the written questions proposed in the programming part should be answered in the same file as the theory questions.

Problem 1: Math Review (20 + 14 + 6 + 10 = 50 points)

Throughout this course, we'll be constantly referencing concepts from linear algebra (e.g. vectors and matrices), probability (e.g. distributions and conditional probability), and calculus (e.g. partial derivatives). The following questions cover some of the core fundamentals that we expect you to be familiar with from the start. If you find yourself struggling with much of this section, you should ask the course staff whether this course is appropriate for you given your background.

1.1 Linear Algebra

(a) (10 points) Provide answers to the following operations or write "invalid" if not possible:

(i) $\begin{bmatrix} -7 & 4 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} -4 \\ 5 \end{bmatrix}$

(ii) $\begin{bmatrix} -4 & 5 \\ 0 & 8 \end{bmatrix} \begin{bmatrix} 7 & -2 \end{bmatrix}$

(iii) $\begin{bmatrix} 2 & -1 & 7 \\ 4 & 0 & 2 \end{bmatrix} \begin{bmatrix} -2 & 3 & -1 \\ 1 & 2 & 0 \end{bmatrix}^\intercal$

(iv) $\begin{bmatrix} 2 & 3 & -1 \end{bmatrix}^\intercal \begin{bmatrix} 5 & 6 \end{bmatrix}^\intercal$

(v) $\begin{bmatrix} 5 & 0 & -1 \end{bmatrix}^\intercal \begin{bmatrix} 2 & 3 \end{bmatrix}$

(b) (10 points) Suppose that we're given two matrices: $\mathbf{w} \in \mathbb{R}^{D \times H}$ and $\mathbf{x} \in \mathbb{R}^{D \times N}$, where $H \neq D \neq N$. What are the output dimension of the following expressions? Write "Invalid" if they cannot be computed.

(i) $\mathbf{x}^\intercal \mathbf{x}$

(ii) $\mathbf{x}\mathbf{x}^\intercal$

(iii) $\mathbf{w}^\intercal \mathbf{x}$

(iv) $\mathbf{x}\mathbf{w}^\intercal$

(v) $\mathbf{w}^\intercal \mathbf{x}\mathbf{x}^\intercal \mathbf{w}$

1.2 Expectation and entropy

(a) (2 points) Given a categorical distribution $P(X)$ as follows, compute the value of $\mathbb{E}[X]$.

x	1	2	3
$P(X = x)$	0.5	0.2	0.3

(b) (2 points) Given a uniform distribution $P(X = x) = \frac{1}{m}, \forall x \in \{1, 2, \ldots, m\}$, compute the value of $\mathbb{E}[X]$.

(c) (2 points) The entropy of a discrete random variable X is defined as (use base e for all log operations unless otherwise specified):

$$H(X) = -\sum_{x \in X} P(x) \log P(x).$$

For the categorical distribution $P(X)$ in problem (a) above, compute the entropy of the distribution.

(d) (2 points) Compute the entropy of the uniform distribution $P(X = x) = \frac{1}{m}, \forall x \in \{1, 2, \ldots, m\}$.

(e) (6 points) Consider the entropy of the joint distribution $P(X, Y)$:

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} P(x, y) \log P(x, y)$$

How does this entropy relate to H(X) and H(Y), (i.e. the entropies of the marginal distributions) when X and Y are independent?

1.3 Bayes' Rule

Suppose that you have two coins in your pocket, one fair coin: $P(H) = P(T) = 0.5$, and one unfair coin: $P(H) = 0.8$ and $P(T) = 0.2$. Suppose that you take one coin from your pocket (each coin equally likely) and flip it.

(a) (2 points) What's the probability that it turns up heads (H)?

(b) (4 points) Suppose that it comes up heads. Given this fact, what's the probability that we picked the unfair coin?

1.4 Calculus

Suppose that we define 4 functions:

$$f(x) = 7x - 2$$

$$h(x) = \frac{f(x)}{2} + 2$$

$$\mathcal{L}(x) = -\log(h(x)^2)$$

$$S(x) = \frac{1}{1 + e^{-x}}$$

(a) (2 points) Compute $\frac{df}{dx}$.

(b) (2 points) Compute $\frac{dh}{dx}$.

(c) (2 points) Compute $\frac{d\mathcal{L}}{dx}$.
(d) (4 points) Show that $\frac{dS}{dx} = S(x)(1 - S(x))$.

Problem 2: Python review (25 + 25 = 50 points)

Throughout this course, all programming assignments will be written in Google colab using Python. In this section of the assignment, we'll review two libraries that are very commonly used in different machine learning applications. Much like Problem 1, if you find yourself struggling with this section, please talk to the course staff to see if this course is appropriate for you at this time.

To get started, open this Colab notebook. For all the coding parts below, **you should not need to create any new files or notebooks**. The notebook has sections where you can fill in the code for all subproblems. Feel free to add and delete arguments in function signatures, but be careful that you might need to change them in function calls which are already present in the notebook.

2.1 Programming with NumPy (25 points)

In this problem, we're going to compute the Euclidean distance matrix, given two matrices of vectors. Specifically, given $X \in \mathbb{R}^{N\times K}$ and $Z \in \mathbb{R}^{M\times K}$, we would like to compute a matrix D such that the entry $d_{i,j}$ at row i, column j is

$$d_{i,j} = \sqrt{(x_i - z_j)^2}$$

where x_i is the i^{th} row of matrix X and z_j is the j^{th} row of matrix Z.
Implement this function in the linked notebook. Note that you may not use any loops or SciPy or NumPy distance functions.

2.2 Using MatPlotLib (25 points)

In this problem, we're going to visualize underfitting and overfitting to a dataset using `matplotlib`. We trained two different models on the training dataset and stored the values each model predicted on the development dataset. The data is present within the `data` folder:

- `data/samples.train.json`: A list of examples the models have been trained on
- `data/true_values.dev.json` : The true (x, y) values of the function for all values of the development set.
- `data/model1_overfit.dev.json` : The (x, y) predictions of the first (overfit) model for all values of the development set.
- `data/model2_underfit.dev.json` : The (x, y) predictions of the second (underfit) model for all values of the development set.

Build two graphs of these models using matplotlib. Specifically, both graphs should:

- Plot training data displayed as a scatter plot. Color the points blue and label them "Samples"
- Plot the true (x, y) values as a line plot. Color it orange and label it "True"
- Set the x and y limits to be (0, 1) and (-1, 3) respectively.
- Add a descriptive title
- Add descriptive x-axis and y-axis labels
- Include a legend

One graph should plot the predictions from the overfitting model and the second graph should plot the predictions from the underfitting model. Both model curves should be colored green and labeled appropriately.