CS 601.471/671 NLP: Self-supervised Models

Homework 2: Word Representation + Linear Classification

For homework deadline, check the calendar on the course website*

Name: _____

Collaborators, if any: _____

Sources used for your homework, if any: _____

This assignment is designed to provide a deeper understanding of linear algebra by incorporating more reviews and focusing on specific topics that were discussed in class. One of the key areas of focus will be on *distance metrics* and their properties. Distance metrics are crucial for measuring the distance between embedding elements in vector space and are a fundamental part of our work. Additionally, we will review several other issues related to *Word2Vec* and *Softmax* that were raised during class discussions. The programming aspect of the assignment will involve building a sentiment classifier using different representational choices, providing an opportunity to apply concepts learned in class in a hands-on manner. Overall, this assignment aims to build a stronger understanding of everything learned thus far.

Homework goals: After completing this homework, you should:

- be familiar with a formal definition of metric functions, popular choices of metrics, and their connections.
- have a good grasp of the Word2Vec algorithm discussed in our class.
- be comfortable with the Softmax function.
- know how to develop your own classifier!

How to hand in your written work: Via Gradescope as before.

1 Linear Algebra Recap

1.1 Gradients

Consider the following scalar-valued function:

$$f(x, y, z) = x^2 y + \sin(z + 6y).$$

1. Compute partial derivatives with respect to x, y and z.
   Answer: *TBD*

2. We can consider f to be a function that takes a vector $\theta \in \mathbb{R}^3$ as input, where $\theta = [x, y, z]^\top$. Write the gradient as a vector and evaluate it at $\theta = [3, \pi/2, 0]^\top$.
   Answer: *TBD*

1.2 Jacobian

Consider the following vector function from $\mathbb{R}^3$ to $\mathbb{R}^3$:

$$\mathbf{f}(\boldsymbol{\theta} = [x_1, x_2, x_3]) = \begin{cases} \sin(x_1 x_2 x_3) \\ \cos(x_2 + x_3) \\ \exp(-\frac{1}{2} x_3^2) \end{cases}$$

*https://self-supervised.cs.jhu.edu/sp2023/

1. What is the Jacobian matrix* of $\mathbf{f}(\boldsymbol{\theta})$?
   Answer: *TBD*

2. Evaluate the Jacobian matrix of $\mathbf{f}(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = [1, \pi, 0]$.
   Answer: *TBD*

2 Meaning Representation and Word Embeddings

2.1 Embeddings

1. Using the class lecture, explain what "semantics" is (one sentence). Then discuss the two takes on semantics and their pros/cons (no more than 5 sentences).
   Answer: *TBD*

2. Explain what one-hot vectors are and what are two difficulties with using them (no more than 5 sentences).
   Answer: *TBD*

3. Explain John Firth's thesis on semantics and how it connects to Word2Vec (no more than 3 sentences).
   Answer: *TBD*

4. Explain how Word2Vec fit the definition of self-supervised algorithms from lecture 1 (no more than 2 sentences).
   Answer: *TBD*

5. Remember the distribution defined for "center" word i and "outside" word o that Word2Vec maximizes:

$$\mathrm{P}\left(o|c\right) = \frac{\exp(u_o \cdot v_c)}{\sum_{x \in V} \exp(u_x \cdot v_c)}$$

   Provide an intuitive explanation of the effect of maximizing the above distribution (1 sentence).
   Answer: *TBD*

6. Discuss how the above function $\mathrm{P}\left(o|c\right)$ leads to the Word2Vec objective function:

$$J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le i \le m \\ j \neq 0}} \log \mathrm{P}\left(w_{t+j}|w_j;\theta\right).$$

   Answer: *TBD*

7. How many learnable parameters θ are there in $J(\theta)$?
   Answer: *TBD*

8. What is the computational complexity of computing a gradient with respect to each parameter using the whole data?
   Answer: *TBD*

9. What is the relationship between the dot product of two word vectors in the skip-gram model and the cosine similarity? For a pair of words with similar semantics, why may the cosine similarity of their word vectors (trained by the skip-gram model) be high?
   Answer: *TBD*

2.2 Challenges in representing language compositionality

A simple way to compute a representation for a phrase s is to add up the representations of the words in that phrase: $\mathrm{repr}(s) = \sum_{w \in s} v_w$, where $w \in s$ are the word in s and v_w is the embedding for word w.

*https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

1. Now, consider sentiment analysis on a phrase in which the predicted sentiments are

$$f(s;\theta) = \theta \cdot \text{repr}(s),$$

for some choice of parameters θ. Prove that in such a model, the following inequality cannot hold for any choice of θ:

$$f(\texttt{good};\theta) > f(\texttt{not good};\theta)$$
$$f(\texttt{bad};\theta) < f(\texttt{not bad};\theta)$$

Thereby, showing the inadequacy of this model in capturing negations.[†]
Answer: *TBD*

2. **Extra Credit:** Construct another example of a pair of inequalities similar to the ones above that cannot both hold.
Answer: *TBD*

3. **Extra Credit:** Consider a slight modification to the previous predictive model:

$$f(s;\theta) = \theta \cdot \text{ReLU}\big(\text{repr}(s)\big),$$

where ReLU (rectified linear function) is defined as:

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$

Given this choice of predictive function, show that it is possible to satisfy the above inequalities for some choice of θ. **Hint:** Show there exists parameters θ and word embeddings $v_{\texttt{good}}$, $v_{\texttt{bad}}$ and $v_{\texttt{not}}$ that the inequalities are satisfied.
Answer: *TBD*

3 Distance Metrics

3.1 Metric properties: Extra Credit:

Definition 3.1. *The function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is considered a **distance function** or **metric** if it satisfies the following three axioms:*

1. ***positivity:*** *the distance between any two points is always non-negative: $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = 0 \Leftrightarrow \boldsymbol{x}_1 = \boldsymbol{x}_2 \;\; \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$.*
2. ***symmetry:*** *$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = d(\boldsymbol{x}_2, \boldsymbol{x}_1) \;\; \forall x_1, x_2 \in \mathbb{R}^d$.*
3. ***triangle inequality:*** *$d(\boldsymbol{x}_1, \boldsymbol{x}_2) + d(\boldsymbol{x}_2, \boldsymbol{x}_3) \geq d(\boldsymbol{x}_1, \boldsymbol{x}_3) \;\; \forall \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \in \mathbb{R}^d$.*

Given this definition, which of the following is a metric function on $\mathbb{R}^n$? Assume that $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$ and $\mathbf{y} = [y_1, \ldots, y_d] \in \mathbb{R}^d$.

1. (euclidean) $d_1(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ Answer: *TBD*
2. (ℓ_0 norm) $d_2(\mathbf{x}, \mathbf{y}) = \sum |x_i - y_i|$ Answer: *TBD*
3. (ℓ_∞ norm) $d_3(\mathbf{x}, \mathbf{y}) = \max_{i=1,\ldots,d} |x_i - y_i|$ Answer: *TBD*
4. (dot product) $d_4(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ Answer: *TBD*
5. (cosine) $d_5(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$ Answer: *TBD*

[†]Question credit: "Introduction to Natural Language Processing" by J. Eisenstein.

3.2 Cosine distance

1. Prove that doubling the length of a vector $\mathbf{x}$ does not change its cosine similarity from any other vector $\mathbf{y}$, i.e. prove that $d_5(2\mathbf{x}, \mathbf{y}) = d_5(\mathbf{x}, \mathbf{y})$.
   Answer: *TBD*

2. Prove that, if $\|\mathbf{x}\| = \|\mathbf{y}\| = 1$, the cosine distance can be expressed in terms of Euclidean distance in the following form:
   $$d_5(\mathbf{x}, \mathbf{y}) = 1 - \frac{d_1(\mathbf{x}, \mathbf{y})}{2}$$
   Answer: *TBD*

4 Softmax function

Remember the Softmax functions from the class:

$$\text{Softmax: } \sigma(\mathbf{z}) \triangleq [\sigma(\mathbf{z})_1, \ldots, \sigma(\mathbf{z})_K] \text{ s.t. } \sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K)$$

1. Prove that Softmax is invariant to constant offsets in the input, i.e., for any input vector $\mathbf{z}$ and any constant c,
   $$\sigma(\mathbf{z}) = \sigma(\mathbf{z} + c)$$
   Pro tip: We make use of this property in practice to increase the numerical stability of our models. Specifically, using $c = -\max_{i \in \{1 \ldots K\}} z_i$, i.e. subtracting its maximum element from all elements of $\mathbf{z}$ would prevent numerical instability due to large values.
   Answer: *TBD*

2. Softmax maintains the relative order of the elements in $\mathbf{z}$. In particular, show that the largest index is intact after applying Softmax:
   $$\arg\max_{i \in \{1 \ldots K\}} z_i = \arg\max_{i \in \{1 \ldots K\}} \sigma(\mathbf{z})_i$$
   Answer: *TBD*

3. Define the Sigmoid function as follows:
   $$\text{Sigmoid: } S(x) \triangleq \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} = 1 - S(-x).$$
   Prove that Softmax is equivalent to the Sigmoid functions when the number of possible labels is two: $K = 2$.
   Answer: *TBD*

4. **Extra Credit:** Next, let's extend (1) to prove the following inequality:
   $$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \geq \prod_{\substack{i=1 \\ j \neq i}}^{n} \frac{1}{1 + e^{-(z_i - z_j)}} = \prod_{\substack{i=1 \\ j \neq i}}^{n} S(z_i - z_j)$$
   Hint: Use the following inequality $(1 + \sum_i \alpha_i) \leq \prod_i (1 + \alpha_i)$ where each $\alpha_i \geq 0$.
   Answer: *TBD*

5 Programming

See the course website for the link to Google Colab. Colab Link