CSC 396 - Introduction to Deep Learning w/ Applications in Natural Language Processing
Jose Santiago Campa Morales
Homework #4
November 23, 2025

Setup (10 points)
https://github.com/jscm1607/csc396-hw4-jscm1607
Text files were not included in the repository because of storage constraints. The notebook includes the solution to Problem #1 and Problem #2. The solution to Problem #3 is in a separate pdf file.

Problem 1 (60 points)
- Time taken: A couple of hours

Problem 2 (50 points)
- Time taken: 45-60 minutes

Problem 3 (30 points)
Consider the simplified self-attention algorithm below, where the only difference from the original algorithm is that it uses a simpler formula to normalize the attention weights a_{ij} in step (b) (instead of the original softmax). Formally, given the query, key, and value vectors for all words in the input text (i.e., $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$ for word w_i), the self attention algorithm operates as follows:

(a) For each pair of words, w_i and w_j, compute the attention weight a_{ij} using the $\mathbf{q}_i$ and $\mathbf{k}_j$ vectors. In particular:
  (i) Initialize the attention weights a_{ij} with the product of the corresponding query and key vectors: $a_{ij} = \mathbf{q}_i \cdot \mathbf{k}_j$
  (ii) Divide the above values by the square root of the length of the key vector: $a_{ij} = a_{ij} / \sqrt{|\mathbf{k}_1|}$ (all vectors have the same size, so we arbitrarily use $\mathbf{k}_1$ here).

(b) For each word w_i, normalize a_{ij} by dividing it by the sum of the elements in $\mathbf{a}_i$: $a_{ij} = \frac{a_{ij}}{\sum_k a_{ik}}$.

(c) For each word w_i, multiply the above attention weights with the corresponding value vectors, for all words w_j. Then sum up all these weighted vectors to produce the output vector for w_i: $\mathbf{z}_i = \sum_j a_{ij} \cdot \mathbf{v}_j$.

Using the algorithm above, compute the $\mathbf{z}_1$ embedding for the token *bagel* in the following text with three tokens: *bagel with cheese*. Assume no other tokenization takes place. Use the following values for the $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$ vectors (indexes start at 1):

q_1 = [1, 2, 3]	k_1 = [1, 1, 1]	v_1 = [2, 0, 1]
q_2 = [2, 3, 2]	k_2 = [0, 0, 0]	v_2 = [3, 0, 0]
q_3 = [5, 6, 7]	k_3 = [2, 2, 0]	v_3 = [1, 2, 2]

Round $\sqrt{|\mathbf{k}_1|}$ to 2.

For (a), let's compute the dot products of each query-key vector pair. Since *bagel* is the first token, we will compute all keys for query a_{1j}:

$$a_{11} = q_1 \cdot k_1 = [1,2,3] \cdot [1,1,1] = 1 * 1 + 2 * 1 + 3 * 1 = 1 + 2 + 3 = 6$$
$$a_{12} = q_1 \cdot k_2 = [1,2,3] \cdot [0,0,0] = 1 * 0 + 2 * 0 + 3 * 0 = 0 + 0 + 0 = 0$$
$$a_{13} = q_1 \cdot k_3 = [1,2,3] \cdot [2,2,0] = 1 * 2 + 2 * 2 + 3 * 0 = 2 + 4 + 0 = 6$$

Next, we divide our attention weights by $\sqrt{|\mathbf{k}_1|}$, which we rounded to 2.

$$a_{11} = a_{11} / \sqrt{|\mathbf{k}_1|} = 6/2 = 3$$
$$a_{12} = a_{12} / \sqrt{|\mathbf{k}_1|} = 0/2 = 0$$
$$a_{13} = a_{13} / \sqrt{|\mathbf{k}_1|} = 6/2 = 3$$

For (b), we normalize a_{1j} by dividing it by the sum of the elements in $\mathbf{a}_i$.

$$\sum_k a_{ik} = a_{11} + a_{12} + a_{13} = 3 + 0 + 3 = 6$$
$$a_{11} = \frac{3}{6} = 0.5$$
$$a_{12} = \frac{0}{6} = 0$$
$$a_{13} = \frac{3}{6} = 0.5$$

For (c), we multiply the attention weights with their value vectors.

$$a_{11} \cdot \mathbf{v}_1 = 0.5 \cdot [2,0,1] = [1,0,0.5]$$
$$a_{12} \cdot \mathbf{v}_2 = 0 \cdot [3,0,0] = [0,0,0]$$
$$a_{13} \cdot \mathbf{v}_3 = 0.5 \cdot [1,2,2] = [0.5,1,1]$$

$$\mathbf{z}_1 = \sum_j a_{ij} \cdot \mathbf{v}_j = a_{11} \cdot \mathbf{v}_1 + a_{12} \cdot \mathbf{v}_2 + a_{13} \cdot \mathbf{v}_3 = [1,0,0.5] + [0,0,0] + [0.5,1,1] = [1.5,1,1.5]$$
$$\mathbf{z}_1 = [1.5,1,1.5]$$

- Time taken: 15-30 minutes (following slides example)