1. Attention exploration (20 points)

Multi-head self-attention is the core modeling component of Transformers. In this question, we'll get some practice working with the self-attention equations, and motivate why multi-headed self-attention can be preferable to single-headed self-attention.

Recall that attention can be viewed as an operation on a *query* vector $q \in \mathbb{R}^d$, a set of *value* vectors $\{v_1, \dots, v_n\}, v_i \in \mathbb{R}^d$, and a set of *key* vectors $\{k_1, \dots, k_n\}, k_i \in \mathbb{R}^d$, specified as follows:

$$c = \sum_{i=1}^{n} v_i \alpha_i \tag{1}$$

$$\alpha_i = \frac{\exp(k_i^\top q)}{\sum_{j=1}^{n} \exp(k_j^\top q)} \tag{2}$$

with $alpha = \{\alpha_1, \dots, \alpha_n\}$ termed the "attention weights". Observe that the output $c \in \mathbb{R}^d$ is an average over the value vectors weighted with respect to α.

(a) (5 points) **Copying in attention.** One advantage of attention is that it's particularly easy to "copy" a value vector to the output c. In this problem, we'll motivate why this is the case.

i. (1 point) **Explain** why α can be interpreted as a categorical probability distribution.

ii. (2 points) The distribution α is typically relatively "diffuse"; the probability mass is spread out between many different α_i. However, this is not always the case. **Describe** (in one sentence) under what conditions the categorical distribution α puts almost all of its weight on some α_j, where $j \in \{1, \dots, n\}$ (i.e. $\alpha_j \gg \sum_{i \neq j} \alpha_i$). What must be true about the query q and/or the keys $\{k_1, \dots, k_n\}$?

iii. (1 point) Under the conditions you gave in (ii), **describe** the output c.

iv. (1 point) **Explain** (in two sentences or fewer) what your answer to (ii) and (iii) means intuitively.

i. Let's check the prerequisites for a categorical probability distribution

1. There are n alpha same as n categories
2. Alpha(i) >= 0 because softmax output is between 0 and 1
3. The sum of alpha = 1 because softmax sum is 1

ii. $\alpha_j \gg \sum_{i \neq j} \alpha_i \Rightarrow \forall i \ \alpha_j \gg \alpha_i \Rightarrow k_j^T q \gg k_i^T q \Rightarrow \alpha_j \approx 1$

this means that kj is much closer to q then to ki

iii. $C = \sum_{i=1}^{n} V_i \alpha_i , \quad C \approx V_j$

iv. One key is almost identical to the given query, the attention weight will be put almost exclusively on its value. Therefore, the output equals that value and make as feel like we copied the key value.

(b) (7 points) **An average of two.** Instead of focusing on just one vector v_j, a Transformer model might want to incorporate information from *multiple* source vectors. Consider the case where we instead want to incorporate information from **two** vectors v_a and v_b, with corresponding key vectors k_a and k_b.

i. (3 points) How should we combine two d-dimensional vectors v_a, v_b into one output vector c in a way that preserves information from both vectors? In machine learning, one common way to do so is to take the average: $c = \frac{1}{2}(v_a + v_b)$. It might seem hard to extract information about the original vectors v_a and v_b from the resulting c, but under certain conditions one can do so. In this problem, we'll see why this is the case.

Suppose that although we don't know v_a or v_b, we do know that v_a lies in a subspace A formed by the m basis vectors $\{a_1, a_2, \ldots, a_m\}$, while v_b lies in a subspace B formed by the p basis vectors $\{b_1, b_2, \ldots, b_p\}$. (This means that any v_a can be expressed as a linear combination of its basis vectors, as can v_b. All basis vectors have norm 1 and are orthogonal to each other.) Additionally, suppose that the two subspaces are orthogonal; i.e. $a_j^\top b_k = 0$ for all j, k.
Using the basis vectors $\{a_1, a_2, \ldots, a_m\}$, construct a matrix M such that for arbitrary vectors $v_a \in A$ and $v_b \in B$, we can use M to extract v_a from the sum vector $s = v_a + v_b$. In other words, we want to construct M such that for any v_a, v_b, $Ms = v_a$. Show that $Ms = v_a$ holds for your M.
Hint: Given that the vectors $\{a_1, a_2, \ldots, a_m\}$ are both *orthogonal* and *form a basis* for v_a, we know that there exist some $c_1, c_2, \ldots, c_m$ such that $v_a = c_1 a_1 + c_2 a_2 + \cdots + c_m a_m$. Can you create a vector of these weights c?

ii. (4 points) As before, let v_a and v_b be two value vectors corresponding to key vectors k_a and k_b, respectively. Assume that (1) all key vectors are orthogonal, so $k_i^\top k_j = 0$ for all $i \neq j$; and (2) all key vectors have norm 1.[1] **Find an expression** for a query vector q such that $c \approx \frac{1}{2}(v_a + v_b)$, and justify your answer. [2]

(b) i. $V_a = c_1 a_1 + c_2 a_2 + \ldots + c_m a_m = AC$

$V_b = d_1 b_1 + d_2 b_2 + \ldots + d_p b_p = BD$

<u>let's find M:</u>

$$V_a = Ms$$
$$V_a = M(V_a + V_b)$$
$$V_a = M(AC + BD)$$
$$AC = M \cdot (AC + BD)$$

(1) $MAC = AC$ (2) $MBD = 0$

$M = AA^T$

(1) $A \overset{I}{\cancel{A^T A}} c = Ac$ (2) $A \overset{0}{\cancel{A^T B}} D = 0$

ii. $c \approx \frac{1}{2}(v_a + \frac{1}{2} v_b) \Rightarrow \alpha_a \approx \alpha_b \gg \alpha_i \ \forall i \neq a, b \Rightarrow$

$k_a^T q \approx k_b^T q \gg k_i^T q \ \forall i \neq a, b \Rightarrow q = \lambda (k_a + k_b)$

(1) $k_a^T \cdot q > 0$ (2) $k_b^T q > 0$ (3) $k_i^T q = 0$

(1) $k_a^T \cdot \lambda (k_a + k_b) = \lambda I + \overset{\text{orthogonal orthogonal}}{\cancel{k_a^T k_b}}$

(2) Same as (1)

(3) $k_i \lambda (k_a + k_b) = \lambda (\overset{\text{orthogonal}}{\cancel{k_i k_a}} + \overset{\text{orthogonal}}{\cancel{k_i k_b}})$

(c) (5 points) **Drawbacks of single-headed attention:** In the previous part, we saw how it was *possible* for a single-headed attention to focus equally on two values. The same concept could easily be extended to any subset of values. In this question we'll see why it's not a *practical* solution. Consider a set of key vectors $\{k_1, \dots, k_n\}$ that are now randomly sampled, $k_i \sim \mathcal{N}(\mu_i, \Sigma_i)$, where the means $\mu_i \in \mathbb{R}^d$ are known to you, but the covariances Σ_i are unknown. Further, assume that the means μ_i are all perpendicular; $\mu_i^\top \mu_j = 0$ if $i \neq j$, and unit norm, $\|\mu_i\| = 1$.

i. (2 points) Assume that the covariance matrices are $\Sigma_i = \alpha I, \forall i \in \{1, 2, \dots, n\}$, for vanishingly small α. Design a query q in terms of the μ_i such that as before, $c \approx \frac{1}{2}(v_a + v_b)$, and provide a brief argument as to why it works.

ii. (3 points) Though single-headed attention is resistant to small perturbations in the keys, some types of larger perturbations may pose a bigger issue. Specifically, in some cases, one key vector k_a may be larger or smaller in norm than the others, while still pointing in the same direction as μ_a. As an example, let us consider a covariance for item a as $\Sigma_a = \alpha I + \frac{1}{2}(\mu_a \mu_a^\top)$ for vanishingly small α (as shown in figure 1). This causes k_a to point in roughly the same direction as μ_a, but with large variances in magnitude. Further, let $\Sigma_i = \alpha I$ for all $i \neq a$.

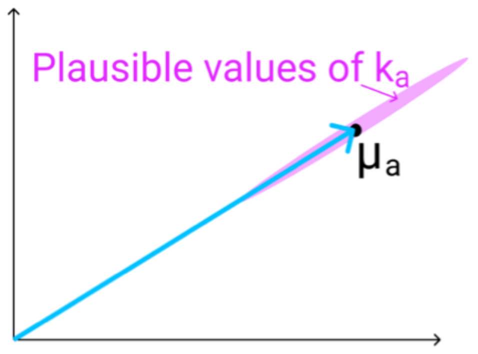

Figure 1: The vector μ_a (shown here in 2D as an example), with the range of possible values of k_a shown in red. As mentioned previously, k_a points in roughly the same direction as μ_a, but may have larger or smaller magnitude.

When you sample $\{k_1, \dots, k_n\}$ multiple times, and use the q vector that you defined in part i., what do you expect the vector c will look like qualitatively for different samples? Think about how it differs from part (i) and how c's variance would be affected.

(c) i. $k_i \sim \mathcal{N}(\mu_i, \Sigma_i)$

$\Sigma_i = \alpha I$ for vanishly small $\alpha \Rightarrow \Sigma_i \approx 0$

$k_i \approx \mu_i$

$\Rightarrow q = \lambda(\mu_a + \mu_b)$

(b.ii) $q = \lambda(k_a + k_b)$

ii. $k_a \in \mathcal{N}(\mu_a, \alpha I + \frac{1}{2}\mu_a\mu_a^T)$

$\alpha I \approx 0$ α is vanishingly small

$\frac{1}{2}\mu_a\mu_a^T = \frac{1}{2}$ $\|\mu_a\| = 1$

$k_a \in \mathcal{N}(\mu_a, \frac{1}{2}) \Rightarrow k_a \approx \epsilon \mu_a$ $\epsilon \in \mathcal{N}(1, 0.5)$

$$\alpha_a = \frac{e^{k_a^T(\lambda(\epsilon\mu_a + \mu_b))}}{e^{k_a^T(\lambda(\epsilon\mu_a + \mu_b))} + e^{k_b^T(\lambda(\epsilon\mu_a + \mu_b))}} = \frac{e^{\epsilon\lambda}}{e^{\epsilon\lambda} + e^{\lambda}}$$

$$\alpha_b = \frac{e^{\lambda}}{e^{\epsilon\lambda} + e^{\lambda}}$$

When ϵ is close to min(0.5)

$\alpha_a \approx 0$ $\alpha_b \approx 1$

When ϵ is close to max(1.5)

$\alpha_a \approx 1$ $\alpha_b \approx 0$

Therefore c will oscillate between Va and Vb when epsilon changes.

(d) (3 points) **Benefits of multi-headed attention:** Now we'll see some of the power of multi-headed attention. We'll consider a simple version of multi-headed attention which is identical to single-headed self-attention as we've presented it in this homework, except two query vectors (q_1 and q_2) are defined, which leads to a pair of vectors (c_1 and c_2), each the output of single-headed attention given its respective query vector. The final output of the multi-headed attention is their average, $\frac{1}{2}(c_1+c_2)$. As in question 1(c), consider a set of key vectors $\{k_1, \ldots, k_n\}$ that are randomly sampled, $k_i \sim \mathcal{N}(\mu_i, \Sigma_i)$, where the means μ_i are known to you, but the covariances Σ_i are unknown. Also as before, assume that the means μ_i are mutually orthogonal; $\mu_i^\top \mu_j = 0$ if $i \neq j$, and unit norm, $\|\mu_i\| = 1$.

i. (1 point) Assume that the covariance matrices are $\Sigma_i = \alpha I$, for vanishingly small α. Design q_1 and q_2 such that c is approximately equal to $\frac{1}{2}(v_a + v_b)$. Note that q_1 and q_2 should have different expressions.

ii. (2 points) Assume that the covariance matrices are $\Sigma_a = \alpha I + \frac{1}{2}(\mu_a \mu_a^\top)$ for vanishingly small α, and $\Sigma_i = \alpha I$ for all $i \neq a$. Take the query vectors q_1 and q_2 that you designed in part i. What, qualitatively, do you expect the output c to look like across different samples of the key vectors? Explain briefly in terms of variance in c_1 and $c2$. You can ignore cases in which $k_a^\top q_i < 0$.

(d) i. let's design q_1 so $c_1 = V_a$ and q_2 so $c_2 = V_b$

then we will get $c = \frac{1}{2}c_1 + \frac{1}{2}c_2 = \frac{1}{2}(V_a + V_b)$

$$C_1 = \sum_{i=1}^{n} V_i \alpha_i \qquad \alpha_a \approx 1 \;,\; \alpha_{i \neq a} \approx 0$$

$$\alpha_a = \frac{e^{k_a^\top q_1}}{\sum_{i=1}^{n} e^{k_i^\top q_1}} \approx 1 \Rightarrow k_a^\top q_1 \gg 0 \Rightarrow q_1 = \lambda k_a$$

$$\Downarrow$$

$$k_i \sim N(\mu_i, \alpha I) \Rightarrow k_i \approx \mu_i \Rightarrow q_1 = \lambda \mu_a$$

$$q_2 = \beta \mu_b$$

ii. (Cii) $k_a \approx \epsilon \mu_a \qquad \epsilon \in N(1, 0.5)$

$$\alpha_a = \frac{e^{\epsilon \mu_a \lambda \mu_a}}{e^{\epsilon \mu_a \lambda \mu_a}} = 1 \qquad \alpha_b = 1$$

$\mu_i^\top \mu_j = 0$ if $i \neq j$

$$c_1 = V_a \qquad c_2 = V_b$$

$$c = \frac{1}{2}(V_a + V_b)$$

i. (8 points) We'll score your model as to whether it gets at least 5% accuracy on the test set, which has answers held out.

ii. (2 points) Provide an expression for the time complexity of the Perceiver model and the vanilla model, in terms of number of layers (L), input sequence length (ℓ) and basis bottleneck dimension (m).

i. 28.1%

ii.

$$Y_i = \text{softmax}\left(\frac{\overset{(l,\frac{d}{h})}{(XQ_i)}\overset{(\frac{d}{h},l)}{(XK_i)^\top}}{\sqrt{d/h}}\right)(XV_i)$$

$[l,l]$

$X \in (l, d)$

$Q, K, V \in (d, \frac{d}{h})$

$XQ, XK, XV \in O(l \cdot d^2)$ +

$(XQ)(XK)^T \in O(l^2 \cdot d)$ +

$[(XQ)(XK)^T] \cdot (XV) \in O(l^2 \cdot d)$

$O(ld^2) + O(l^2 d) + O(l^2 d) \in O(l^2 d)$ (1) $l >> d$

$$Y_i^{(L)} = \text{softmax}\left(\frac{\overset{(m,\frac{d}{h})}{(XQ_i)}\overset{(m,\frac{d}{h})}{(Y^{(L-1)}K_i)^\top}}{\sqrt{d/h}}\right)\overset{(m,\frac{d}{h})}{(Y^{(L-1)}V_i)}$$

$[m,m]$

$X \in (m, d)$

$Q, K, V \in (d, \frac{d}{h})$

$Y \in (m, \frac{d}{h})$

$XQ \in \Theta(md^2)$

$Y_{L-1}K, Y_{L-1}V \in \Theta(md^2)$

$[(XQ)(Y_{L-1}K)^T] \in \Theta(m \cdot d \cdot m)$

$[\;] \cdot Y_{L-1}V \in \Theta(m^2 d)$

$\Theta(md^2) + \Theta(md^2) + \Theta(m^2 d) + \Theta(m^2 d) \in \Theta(m^2 d)$ (2) $m >> d$

$2 \cdot \overset{(1)}{\Theta(l^2 d)} + (L-2) \cdot \overset{(2)}{\Theta(m^2 d)} = \Theta(Lm^2 d)$

3. Considerations in pretrained knowledge (5 points)

Please type the answers to these written questions (to make TA lives easier).

(a) (1 point) Succinctly explain why the pretrained (vanilla) model was able to achieve an accuracy of above 10%, whereas the non-pretrained model was not.

(a) The model had seen the correct answer more times in his training, he also had much more context to learn from.

(b) (2 points) Take a look at some of the correct predictions of the pretrain+finetuned vanilla model, as well as some of the errors. We think you'll find that it's impossible to tell, just looking at the output, whether the model *retrieved* the correct birth place, or *made up* an incorrect birth place. Consider the implications of this for user-facing systems that involve pretrained NLP components. Come up with two **distinct** reasons why this model behavior (i.e. unable to tell whether it's retrieved or made up) may cause concern for such applications, and an example for each reason.

(b) 1. Misinformation - if the model can not tell whether it’s retrieved or made up it can provide the user with misleading data. The user can be provided with wrong birthplace of a public figure and may share the information leading to a loss in his credibility.

2. Legal implications - the model can made up roles and legal information that can mislead user to act illegally and break the law.

(c) (2 points) If your model didn’t see a person’s name at pretraining time, and that person was not seen at fine-tuning time either, it is not possible for it to have “learned” where they lived. Yet, your model will produce *something* as a predicted birth place for that person’s name if asked. Concisely describe a strategy your model might take for predicting a birth place for that person’s name, and one reason why this should cause concern for the use of such applications. (You do not need to submit the same answer for 3c as for 3b.)

(c) The model can guess the birthplace from the ethnical source of the name (he had seen that most of the ‘Francois’ has been born in France) this is a solid guess but it is biased - one might call Dylan and born in Israel just because his parents loved this name or they had relocation for work purposes.