Hoorieh Sabzevari – 98412004

NLP – A5 (written)

1) Attention exploration

(a)

i. In Attention, α refers to the attention weights assigned to each input token or element in a sequence. These attention weights are non-negative and sum up to 1, indicating the relative importance of each token in generating the output. Since the attention weights represent a probability distribution over the input tokens, they can be interpreted as a categorical probability distribution. This allows the model to selectively focus on the most relevant parts of the input sequence when generating the output.

ii. The categorical distribution α will put almost all of its weight on some α_j, where $j \in \{1, . . . , n\}$, if the query q is very similar to one of the keys k_i in the set $\{k_1, . . . , k_n\}$. This happens because the corresponding probability in the categorical distribution will be significantly larger than the other probabilities. The degree to which this occurs depends on the similarity between the query and the keys. If the query is dissimilar to all of the keys, then the distribution will be relatively diffuse with the probability mass spread out between many different α_i.

$$k_j^\top q \gg k_i^\top q \qquad \forall i \neq j$$

iii. $\alpha_i > 0\,,\ \ \sum \alpha_i = 1\,,\ \ \alpha_j \gg \sum_{i \neq j} \alpha_i$

$$\alpha_j \approx 1 \ \ so\ that\ \ c = \sum_{i=1}^{n} v_i \alpha_i = v_j \alpha_j = v_j$$

iv. The categorical distribution assigns probabilities to a set of keys. If the query is very similar to one or more of the keys, then the corresponding probability in the distribution will be much higher than the other probabilities, and the key with the highest probability will be selected as the output.

(b)

i. Let A be a matrix formed by concatenating basis vectors $\{a_1, a_2, . . . , a_m\}$, and let B be a matrix formed by concatenating basis vectors $\{b_1, b_2, . . . , b_p\}$. Using these matrices, we can express linear combinations of v_a and v_b as follows:

$$v_a = c_1a_1 + c_2a_2 + \cdot\cdot\cdot + c_ma_m = A_c$$

$$v_b = d_1b_1 + d_2b_2 + \cdot\cdot\cdot + d_pb_p = B_d$$

Our goal is to find a matrix *M* that satisfies two conditions: first, when *M* is multiplied by v_b, the resulting vector is the zero vector; second, when *M* is multiplied by v_a, the resulting vector is the same as v_a (but expressed in the coordinate system of *M*).

$$Ms = v_a$$

$$Mv_a + Mv_b = v_a$$

We can observe that for all j and k, $a_j Tb_k = 0$, $A^TB = 0$. And, since $a_i{}^Ta_j = 0$ whenever $j \neq i$ and since $a_i{}^Ta_j = 1$ whenever $j = i$ because vectors are normalized, $A^TA = I$. If we substitute *M* with A^T, we obtain:

$$A^TA_c + A^TB_d = Ic + 0d = c$$

And we know that in terms of $\mathbb{R}^d$ (not in terms of *A* and *B*), v_a is just a collection of constants *c*. Thus $M = A^T$

ii. From $c \approx 0.5v_a + 0.5v_b$, follows $\alpha_a = \alpha_b = 0.5$ And from (a) we know that this means:

$$k_a^\top q \approx\ k_b^\top q \gg k_i^\top q\ ,\ \forall i \neq a, b\ .$$

Let $k_a{}^Tq = k\ {}_b{}^T\ q = \beta$,

then $\frac{\exp(\beta)}{\sum_{j=1}^{n} \exp(\beta)} = \frac{\exp(\beta)}{n-2+2\exp(\beta)}$ and for β ≫ 0 ⇒ *exp(β)* → ∞ we get ≈ $\frac{\exp(\beta)}{2\exp(\beta)} = \frac{1}{2}$

So $q = \beta(k_a + k_b)$ *with* $\beta \gg 0$.

(c)

i. Given that the variances (represented by diagonal covariance values) for $i \in \{1, 2, ..., n\}$ are extremely small, we can infer that each key vector is in close proximity to its mean vector: $k_i \approx \mu_i$

Since all of the mean vectors are perpendicular to each other, we can simplify the problem to the previous scenario where all keys were also perpendicular to each other. As a result, q can be represented as:

$$q = \beta(\mu_a + \mu_b),\ where\ \beta \gg 0$$

ii. Since $\mu_i^T\mu_i = 1$, ka varies between $(\alpha + 0.5)\mu_a$ and $(\alpha + 1.5)\mu a$. All other k_i , whenever $i \neq a$, almost don't vary at all. Noting that α is vanishingly small:

$$k_a^T q \approx \gamma\mu_a^T\beta(\mu_a + \mu_b) \approx \gamma\beta,\ where\ \beta \gg 0$$

$$k_b^T q \approx \mu_b^T\beta(\mu_a + \mu_b) \approx \beta,\ where\ \beta \gg 0$$

We can now directly solve for coefficients αa and αb, remembering that for large β values $exp(0)$ are insignificant:

$$\alpha_a \approx \frac{\exp(\gamma\beta)}{\exp(\gamma\beta) + \exp(\beta)} \approx \frac{1}{1 + \exp(\beta(1-\gamma))}$$

$$\alpha_b \approx \frac{\exp(\beta)}{\exp(\beta) + \exp(\gamma\beta)} \approx \frac{1}{1 + \exp(\beta(\gamma-1))}$$

Since γ varies between 0.5 and 1.5, and since $\beta \gg 0$, we have that:

$$\alpha_a \approx \frac{1}{1+\infty} \approx 0; \quad \alpha_b \approx \frac{1}{1+0} \approx 1 \quad when\ \gamma = 0.5$$

$$\alpha_a \approx \frac{1}{1+0} \approx 1; \quad \alpha_b \approx \frac{1}{1+\infty} \approx 0 \quad when\ \gamma = 1.5$$

Since $c \approx \alpha_a v_a + \alpha_b v_b$ because other terms are insignificant when β is large, we can see that c oscillates between v_a and v_b:

$$c \approx v_b,\ when\ \gamma \rightarrow 0.5; \qquad c \approx v_a,\ when\ \gamma \rightarrow 1.5$$

(d)

i. With the same assumptions as before, we can design q1 and q2 such that one of them copies v_a and another copies v_b. Since all keys are similar to their means and following the explanation in question (a) iv., we express the queries as:

$$q_1 = \beta\mu_a,\ q2 = \beta\mu_b,\ for\ \beta \gg 0$$

This gives us: $c_1 \approx v_a; \ c_2 \approx v_b$ And since multiheaded attention is just an average of the 2 values, we can see that:

$$c \approx \frac{1}{2}\,(v_a + v_b)$$

ii. With regards to question (c) ii., if we choose $q1 = \beta\mu_a$ and $q2 = \beta\mu_b$,we get that (note that all other key-query dot products will be insignificant):

$$k_a{}^T q \approx \gamma\mu_a{}^T\beta(\mu_a + \mu_b) \approx \gamma\beta, \text{ where } \beta \gg 0$$

$$k_b{}^T q \approx \mu_b{}^T\beta(\mu_a + \mu_b) \approx \beta, \text{ where } \beta \gg 0$$

We can solve for α values (again, note that all other key-query dot products will be insignificant when β is large):

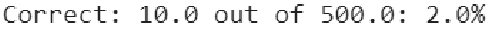
$$\alpha_{a1} \approx \frac{\exp(\gamma\beta)}{\exp(\gamma\beta)} \approx 1; \quad \alpha_{b2} \approx \frac{\exp\,(\beta)}{\exp\,(\beta)} \approx 1$$

Since we can say that $\alpha_{i1} \approx 0$ for any $i \neq a$ and $\alpha_{i2} \approx 0$ for any $i \neq b$ is easy to see that:

$$c_1 \approx v_a,\ c_2 \approx v_b \text{ , So } c \approx \frac{1}{2}\,(v_a + v_b).$$

2) Pretrained Transformer models and knowledge access

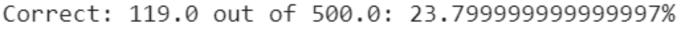
(d)



```
Correct: 10.0 out of 500.0: 2.0%
```

London: 5%

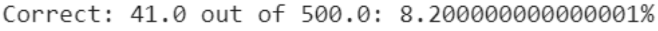
(f)



```
Correct: 119.0 out of 500.0: 23.799999999999997%
```

(g)

i.



```
Correct: 41.0 out of 500.0: 8.200000000000001%
```

ii. The Attention layers play a vital role in the perceiver approach. In case the input dimensionality is denoted as d, the QKV attention operation's complexity is O(d × ℓ). However, using cross-attention, we can project K and V from the input while Q is a projection of a learned latent vector reduced by m. The dimension of this latent vector is m, which is considerably smaller than ℓ. This way, the attention operation's complexity reduces to O(d × m). Similarly, the self-attentions' complexity in the latent transformer blocks also decreases to O(m2). While multi-headed attention has a time complexity of

$O(\ell2d + \ell d2)$, the perceiver model's time complexity is $O(dm + Lm2)$. Here, d refers to the byte array's dimensionality, m represents the latent array's dimensionality, and L denotes the transformer's depth.

3) Considerations in pretrained knowledge

(a) The pre-trained model was able to achieve higher accuracy due to its prior knowledge gained from being trained on a vast dataset. By learning patterns and features during the pre-training process, it became better equipped to generalize on new examples. On the other hand, the non-pretrained model had to start from scratch, lacking initial knowledge and taking longer to learn significant representations, ultimately resulting in lower accuracy.

(b) 1. The potential consequences of this behavior could result in users unknowingly citing inaccurate information in their work. For example, if a user is attempting to identify the birthplace of a famous person and the model fails to provide accurate information, the user may inadvertently provide incorrect information, adversely affecting the quality of their work.

2. The behavior of the model in retrieving inaccurate information has the potential to cause users to unwittingly disseminate false information related to the given subject matter. For instance, if a user relies on the model to retrieve information about a famous person, such as their birthplace, and the retrieved information proves to be false, the user may inadvertently propagate erroneous facts about the individual which others may come to believe. This situation could lead to confusion and disputes within society.

(c) If a name is given to the model, it will attempt to optimize its accuracy by utilizing its parameters in order to retrieve information that is associated with people who possess similar names. This feature is particularly useful in cases where there may be a typographical error in the name provided. However, if the name provided is only similar to one of the names that the model has previously learned about, the retrieved information may not be accurate and could potentially lead to quality and social concerns, as described in 3b.