NLP HW3

Yoav Rabinovich   Ido Shapira
208275735   319021044

Question 1

1.

a. α can be interpreted as a categorical probability distribution because if we view it as a categorical probability distribution with n categories such that each category is I and the probability is α_i then we can see it has all the needed properties:
$\sum_{i=1}^{n} \alpha_i = 1\ , for\ all\ i\ \alpha_i \geq 0.$

b. The categorical distribution α puts almost all of its weight on some αj when the dot product between the query q and a specific key kj is significantly larger than the dot products between the query and all other keys, indicating a strong similarity or match between q and kj compared to the other keys.

c. The output C will be very close to v_j for the j from the last question.

d. In intuitive terms, this means that when the query q closely matches or aligns with a specific key kj compared to the other keys, the resulting output c will be strongly influenced by the vector vj associated with that key. It implies that the model focuses its attention and assigns a higher weight to the key that exhibits a stronger similarity or relevance to the query, resulting in a more pronounced impact on the final output.

2.

a. We will observe M = $\begin{pmatrix} \sum_{j=1}^{m} a_{j1} * a_j \\ \dots \\ \sum_{j=1}^{m} a_{jm} * a_j \\ 0 \\ \dots \\ 0 \end{pmatrix}$.

$$Ms = \begin{pmatrix} \sum_{j=1}^{m} a_{j1} * c_j \\ \dots \\ \sum_{j=1}^{m} a_{jm} * c_j \\ 0 \\ \dots \\ 0 \end{pmatrix} = v_a$$

b. We will observe $q = m(k_a + k_b)\ where\ m\ is\ a\ large\ scalar$.

$$\alpha_a = \frac{\exp(k_a^T q)}{\sum_{j=1}^{n} \exp\,(k_j^T q)} = \frac{\exp(m(k_a^T k_a + k_a^T k_b))}{\sum_{j=1}^{n} \exp\,(m\big(k_j^T k_a + k_j^T k_b\big))} = \frac{\exp(m)}{2\exp(m) + n - 2} \approx \frac{1}{2}$$

$$\alpha_b = \alpha_a \approx \frac{1}{2}$$

$$\alpha_i = 1 - \alpha_a - \alpha_b \approx 0$$

$$c = \sum_{i=1}^{n} v_i \alpha_i \approx \frac{1}{2}(v_a + v_b)$$

3.

a)

We will define $q = m(\mu_a + \mu_b)$.

Since $\Sigma_i = \alpha I, for\ vanishingly\ small\ \alpha$, we can say that $\mu_i \approx k_i$.

The fact that $\forall_{i\neq j}\ \mu_i^T \mu_j = 0$ says that approximately $\forall_{i\neq j}\ k_i^T k_j = 0$ and we already solve this case in question 2b: $q = m(k_a + k_b)\ \rightarrow c \approx \frac{1}{2}(v_a + v_b)$

All in all, $q = m(\mu_a + \mu_b). \rightarrow c \approx \frac{1}{2}(v_a + v_b)$.

b)

Now we have $\Sigma_a = \alpha I + \frac{1}{2}(\mu_a \mu_a^T), \Sigma_{i\neq a} = \alpha I$

As in last question, $\mu_{i\neq a} \approx k_i$

We also know that $\big|\big|\mu_a^T \mu_a\big|\big| = 1\ and\ \Sigma_a\ definition \rightarrow k_a \sim norm\left(\mu_a, \frac{1}{2}\mu_a\right)$, to make the description simpler we will sample $\epsilon \sim norm(1, \frac{1}{2})$ and define $k_a \approx \epsilon * \mu_A$

The dot product between q and $k_i\ for\ i \neq a, b$ will be zero just like in previous question so we need to calculate the following SoftMax participants:

$$k_a^T q = m(k_a^T k_b + k_a^T k_a) = m(0 + \epsilon) = \epsilon m$$
$$k_b^T q = m\big(k_b^T k_b + k_b^T k_a\big) = m(1 + 0) = m$$

$$\alpha_a = \frac{\exp(k_a^T q)}{\sum_{j=1}^{n} \exp\,(k_j^T q)} = \frac{\epsilon m}{\epsilon m + m + n - 2} \approx \frac{\epsilon}{1 + \epsilon}$$
$$\alpha_b = \frac{\exp\big(k_b^T q\big)}{\sum_{j=1}^{n} \exp\,(k_j^T q)} = \frac{m}{\epsilon m + m + n - 2} \approx \frac{1}{1 + \epsilon}$$

Which means $c = \frac{\epsilon v_a}{1+\epsilon} + \frac{v_b}{1+\epsilon} \approx \frac{1}{1+\epsilon}(\epsilon v_a + v_b)$ where $\epsilon \sim norm(1, \frac{1}{2})$.

4.

a)

We will define q1 and q2 as follows:

$$q_1 = m\mu_a$$
$$q_2 = m\mu_b$$

As before $k_i \approx \mu_i$.

$$\underline{Head\#1:}$$

$$\alpha_a = \frac{\exp(k_a^T q_1)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{\exp(mk_a^T k_a)}{\exp(mk_a^T k_a) + n - 1} \approx 1$$
$$\alpha_{i \neq a} = \frac{\exp\left(k_i^T q_1\right)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{\exp\left(k_i^T k_a\right)}{\exp(mk_a^T k_a) + n - 1} \approx 0$$
$$c_1 = 1 * v_a = v_a$$

With similar calculations, we infer $c_2 = v_b$.

$$c = \frac{1}{2}(c_1 + c_2) = \frac{1}{2}(v_a + v_b)$$

b)

As in previous question we will sample $\epsilon \sim norm\left(1, \frac{1}{2}\right)$.

$$k_a \approx \epsilon * \mu_A$$
$$k_i \approx \mu_{i \neq a}$$

$$k_a^T q_1 = m(k_a^T k_a) = \epsilon m$$
$$k_b^T q_2 = m\left(k_b^T k_b\right) = m$$

$$\underline{Head\#1:}$$

$$\alpha_a = \frac{\exp(k_a^T q_1)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{\epsilon m}{\epsilon m + n - 1} \approx 1$$
$$\alpha_{i \neq a} = \frac{\exp\left(k_i^T q_1\right)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{\exp\left(k_i^T k_a\right)}{\exp(mk_a^T k_a) + n - 1} \approx 0$$
$$c_1 = 1 * v_a = v_a$$

$$\underline{Head\#2:}$$

$$\alpha_a = \frac{\exp(k_a^T q_1)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{m}{m + n - 1} \approx 1$$
$$\alpha_{i \neq a} = \frac{\exp\left(k_i^T q_1\right)}{\sum_{j=1}^{n} \exp\left(k_j^T q_1\right)} = \frac{\exp\left(k_i^T k_a\right)}{\exp(mk_a^T k_a) + n - 1} \approx 0$$
$$c_1 = 1 * v_b = v_b$$

All in all, $c = \frac{1}{2}(v_a + v_b)$

Question 2

4. When we evaluate our model that did not do pretrain we get for the dev set:
Correct: 9.0 out of 500.0: 1.7999999999999998%

When we predict London every time we get for the dev set:
Correct: 25.0 out of 500.0: 5.0%

We can see that the model now is very bad. Even worse that a model that give the same prediction for every input.

6. When we evaluate our model that did do pretrain we get for the dev set:
Correct: 90.0 out of 500.0: 18.0%

7. The pretrained vanilla model achieved high accuracy because it had been trained on relevant data, acquiring knowledge of language patterns, grammar, and semantics. This prior training provided a solid foundation for better predictions and responses. In contrast, the non-pretrained model probably did not gain enough understanding of the language semantics and structure from the fine tune data to successfully infer the correct answers from the data.