CS 224N - Assignment 5

Ilya Kachan

July 21, 2021

1 Attention exploration

(a) Query vector q and key vectors k_i must be such that the dot product $k_j^T q$ dominates over all the rest scalar products $k_i^T q$, $i \neq j$.

(b). Put $q = Q(k_a + k_b)$ for sufficiently large $Q > 0$. Then

$$k_i^T q = \left\{ \begin{array}{ll} Q, & i \in \{a, b\} \\ 0, & i \notin \{a, b\} \end{array} \right. , \qquad \alpha_i = \left\{ \begin{array}{ll} \frac{e^Q}{n-2+2e^Q}, & i \in \{a, b\} \\ \frac{1}{n-2+2e^Q}, & i \notin \{a, b\} \end{array} \right. ,$$

Note that as $Q \to \infty$ in case $i \in \{a, b\}$ we obtain $\alpha_i \to 1/2$, while in case $i \notin \{a, b\}$: $\alpha_i \to 0$.

(c) i. Put $q = Q(\mu_a + \mu_b)$. According to the properties of normal distribution: $k_i^T \mu_j = \mu_j^T k_i \sim N(\mu_j^T \mu_i, \mu_j^T \Sigma_i \mu_j) = N(\mu_j^T \mu_i, \alpha I)$. Thus $k_i^T \mu_j \approx \mu_j^T \mu_i$ – non-random value, when α is sufficiently small. This value equals 0 for $i \neq j$ and 1 otherwise. The rest explanation is the same as in the point (b).

ii. Note that $k_a^T \mu_j = \mu_j^T k_a \sim N(\mu_j^T \mu_a, \mu_j^T \Sigma_a \mu_j) = N(\mu_j^T \mu_a, (\alpha + (\mu_j^T \mu_a)^2/2)I)$. Thus, if $j \neq a$, $k_a^T \mu_j \approx 0$ and $k_a^T \mu_a \sim N(\|\mu_a\|^2, (\alpha + \|\mu_a\|^4/2)I)$. Depending on the sampled vector the value of $k_a^T \mu_a$ might be rather big or small. Therefore, the vector c expected to be roughly equal to $pv_a + (1-p)v_b$, where $p \in [0, 1]$ and p depends on the $\|k_a\|$.

(d) i. Analogiously to (c): $q_1 = Q\mu_a$, $q_2 = Q\mu_b$.

ii. I expect the output c to be still approximately equal to $\frac{1}{2}(q_1 + q_2)$. Regardless of how big $\|k_a\|$ is, the component $k_a^T q_1$ still dominates over $k_i^T q_1$, and all the more $k_b^T q_2$ still dominates over $k_i^T q_2$.

(e) i. Let's do some computations:

$$\langle x_1, x_1 \rangle = \|u_b\|^2 + \|u_d\|^2 = 2\beta^2, \quad \langle x_1, x_2 \rangle = 0, \quad \langle x_1, x_3 \rangle = \|u_b\|^2 = \beta^2,$$

$$\langle x_2, x_2 \rangle = \|u_a\|^2 = \beta^2, \quad \langle x_2, x_3 \rangle = 0, \quad \langle x_3, x_3 \rangle = \|u_c\|^2 + \|u_b\|^2 = 2\beta^2.$$

$$\alpha_{11} = \frac{e^{2\beta^2}}{e^{2\beta^2} + 1 + e^{\beta^2}} \to 1, \quad \alpha_{22} = \frac{e^{\beta^2}}{2 + e^{\beta^2}} \to 1, \quad \alpha_{33} = \frac{e^{2\beta^2}}{e^{\beta^2} + 1 + e^{2\beta^2}} \to 1,$$

$$\alpha_{12} = \alpha_{32} = \frac{1}{e^{2\beta^2} + 1 + e^{\beta^2}} \to 0, \quad \alpha_{13} = \alpha_{31} = \frac{e^{\beta^2}}{e^{2\beta^2} + 1 + e^{\beta^2}} \to 0,$$

$$\alpha_{21} = \alpha_{23} = \frac{1}{2 + e^{\beta^2}} \to 0.$$

Hence $c_2 = \sum_{j=1}^{3} \alpha_{2j} v_j = \alpha_{21}(u_d + u_b) + \alpha_{22} u_a + \alpha_{23}(u_c + u_b) \approx u_a$. Adding u_d or u_c to x_2 won't help to approximate u_b with c_2.

ii. First, consider $V = \beta^{-2}(u_b u_b^T - u_c u_c^T)$ and compute

$$v_1 = V(u_d + u_b) = \beta^{-2} u_b (u_b^T u_d) + \beta^{-2} u_b (u_b^T u_b) - \beta^{-2} u_c (u_c^T u_d) - \beta^{-2} u_c (u_c^T u_b) = u_b,$$

$$v_2 = V u_a = \beta^{-2}(u_b (u_b^T u_a) - u_c (u_c^T u_a)) = 0,$$

$$v_3 = V(u_c + u_b) = \beta^{-2} u_b (u_b^T u_c) + \beta^{-2} u_b (u_b^T u_b) - \beta^{-2} u_c (u_c^T u_c) - \beta^{-2} u_c (u_c^T u_b) = u_b - u_c.$$

The ultimate goal is to achieve the following relations:

$$c_1 = \alpha_{11} v_1 + \alpha_{12} v_2 + \alpha_{13} v_3 = \alpha_{11} u_b + \alpha_{13}(u_b - u_c) \approx u_b - u_c,$$

$$c_2 = \alpha_{21} v_1 + \alpha_{22} v_2 + \alpha_{23} v_3 = \alpha_{21} u_b + \alpha_{23}(u_b - u_c) \approx u_b.$$

To achive that, it's sufficient to obtain

$$\alpha_{11} \to 0, \quad \alpha_{12} \to 0, \quad \alpha_{13} \to 1,$$

$$\alpha_{21} \to 1, \quad \alpha_{22} \to 0, \quad \alpha_{23} \to 0,$$

$$\alpha_{31} \to 0, \quad \alpha_{32} \to 1, \quad \alpha_{33} \to 0.$$

In turn, since $\alpha_{ij} = \frac{\exp(k_j^T q_i)}{\sum_{l=1}^{n} \exp(k_l^T q_i)}$, that will be true, if

$$k_1^T q_1 = 0, \quad k_2^T q_1 = 0, \quad k_3^T q_1 = \beta^2,$$

$$k_1^T q_2 = \beta^2, \quad k_2^T q_2 = 0, \quad k_3^T q_2 = 0,$$

$$k_1^T q_3 = 0, \quad k_2^T q_3 = \beta^2, \quad k_3^T q_3 = 0.$$

It's easy to see that these equations will be true, if we put

$$k_1 = u_a, \quad k_2 = u_c, \quad k_3 = u_b,$$

$$q_1 = u_b, \quad q_2 = u_a, \quad q_3 = u_c.$$

So what remains is to choose matrices K, Q such that the equations above are fulfilled. Let's prove that the matrices

$$K = \beta^{-2}(u_c u_a^T + u_a u_d^T + u_b u_c^T),$$

$$Q = \beta^{-2}(u_a u_a^T + u_b u_d^T + u_c u_c^T).$$

will suit. Let's prove that:

$$k_1 = K(u_d + u_b) =$$

$$= \beta^{-2}(u_c(u_a^T u_d) + u_a(u_d^T u_d) + u_b(u_c^T u_d) + u_c(u_a^T u_b) + u_a(u_d^T u_b) + u_b(u_c^T u_b)) = u_a$$

$$k_2 = K u_a = \beta^{-2}(u_c(u_a^T u_a) + u_a(u_d^T u_a) + u_b(u_c^T u_a)) = u_c$$

$$k_3 = K(u_c + u_b) =$$

$$= \beta^{-2}(u_c(u_a^T u_c) + u_a(u_d^T u_c) + u_b(u_c^T u_c) + u_c(u_a^T u_b) + u_a(u_d^T u_b) + u_b(u_c^T u_b)) = u_b$$

$$q_1 = Q(u_d + u_b) =$$

$$= \beta^{-2}(u_a(u_a^T u_d) + u_b(u_d^T u_d) + u_c(u_c^T u_d) + u_a(u_a^T u_b) + u_b(u_d^T u_b) + u_c(u_c^T u_b)) = u_b$$

$$q_2 = Q u_a = \beta^{-2}(u_a(u_a^T u_a) + u_b(u_d^T u_a) + u_c(u_c^T u_a)) = u_a$$

$$q_3 = Q(u_c + u_b) =$$

$$= \beta^{-2}(u_a(u_a^T u_c) + u_b(u_d^T u_c) + u_c(u_c^T u_c) + u_a(u_a^T u_b) + u_b(u_d^T u_b) + u_c(u_c^T u_b)) = u_c.$$

2 Pretrained Transformer models and knowledge access

(d) Vanilla model's accuracy on the dev set is 1,4%.

London baseline model's accuracy on the same dev set is 5%.

(f) Pretrained finetuned vanilla model's accuracy on the dev set is 21,8%.

(g) Synthesizer attention model's accuracy on the dev set is 6,2%.

Synthesizer might not be able to perform at the same level as the ordinary self-attention, since it has less parameters to capture some advanced patterns.

3 Considerations in pretrained knowledge

(a) Pretraining helped a lot because of the span corruption technique used - by randomly noising the input text we've emulated model training on larger ammount of data. So we've pretrained the model on the "large" ammount of data which we couldn't do with the initial task.

(b) 1) Predicting the birplace of the persons previously unseen by the model becomes risky. 2) The people with exactly the same names may live in different places.

(c) The model might look at similar person names (if a person name seem to German, look at German names, for example) and try to guess the birthplace based on the data of the people with similar names. The problem here is that the even the person with German name, for example, might live anywhere, not necessarily in Germany.