Natural Language Processing
Assignment №5

Pooya Kabiri

Department of Computer Science

Iran University of Science and Technology

June, 2021

Contents

1 Attention exploration

a.

For this to happen, the value of $c = \sum_{i=1}^{n} v_i \alpha_i$ must be almost equal to v_j. In order for this, the value of α_j should be ≈ 1 and α_i for i values other than j, should be ≈ 0.

If the value of $k_i^\top q$ for the specific i value (which we call it j), be greater than all other i values, then in the softmax operation, it's attention weight is almost equal to 1 and other values would be close to zero.

Mathematically, the condition is:

$$k_j^\top q \ggg k_i^\top q, i \neq j$$

b.

The requested equation must be $c = \sum_{i=1}^{n} v_i \alpha_i \approx \frac{1}{2}(v_a + v_b)$. In order for this to happen, values of α_i corresponding to v_a and v_b must be $\approx \frac{1}{2}$, and for i values not equal to a and b, α_i must be ≈ 0.

Mathematically, the condition is:

$$c = \frac{1}{2}(v_a + v_b) \Rightarrow \alpha_a \approx \alpha_b \approx 1,\ \alpha_i \approx 0 \ \ i \notin \{a, b\}$$

$$\Rightarrow k_a^\top q \approx k_b^\top q \ggg k_i^\top q \ i \notin \{a, b\}$$

Now, if the query vector is a linear sum of the k_a and k_b vectors, then it is on the dimensions of only k_a and k_b and is orthogonal to all other k_i vectors, so the inner product equals zero. In order to satisfy the greatness condition (so that the value after softmax equals 1 and others equal 0), we can multiply this linear sum with a very large positive value.

So, the ideal value for q is:

$$q = k_a + k_b \xrightarrow{\times M} q = M(k_a + k_b)$$

Where M is a large positive integer.

c.

i.

It is stated that the value of α is vanishingly small, that means that the variance in each normal distribution is almost equal to zero and can be neglected. When this happens, the k_i vectors are almost equal to μ_i, and we know that μ_i vectors are orthogonal mutually, so the solution is identical to part b:

$$q = M(k_a + k_b) \xrightarrow{k_i \approx \mu_i} q = M(\mu_a + \mu_b)$$

Where M is a large positive integer.

ii.

For every sample k_i with $i \neq a$, the q vector is approximately equal to the μ vector and all equations are equivalent to part i. But for k_a :

$$k_a \sim \mathcal{N}(\mu_a, \alpha I + \frac{1}{2}(\mu_a \mu_a^\top)$$

$$\Rightarrow k_a \sim \mathcal{N}(\mu_a, \frac{1}{2}\mu_a^2) \xrightarrow{\|\mu_i\|=1} k_a \sim \mathcal{N}(1, \frac{1}{2})$$

So k_a is almost equal to μ_a but with a little variance which is sampled from $\mathcal{N}(1, \frac{1}{2})$. This is a normal distribution with norm of 1 and variance of $\frac{1}{2}$.

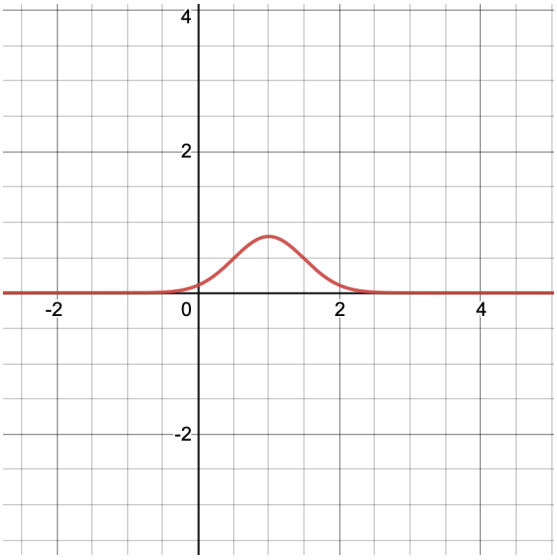

Figure 1: $\mathcal{N}(1, \frac{1}{2})$

As can be seen from the plot, the value oscillates from ≈ 0 to ≈ 2. So we can denote the value of k_a as $\epsilon_a \mu_a$ with ϵ equal to the variance and $\epsilon \sim \mathcal{N}(1, \frac{1}{2})$

Now we assume the same case of part i, and use the q vector we found there. So for attention weights we have:

$$\alpha_i = k_i^\top q \approx \mu_i^\top q \approx 0, \; i \notin \{a, b\}$$

$$\alpha_b = k_b^\top q \approx \mu_b^\top q \approx M \ggg 0$$

$$\alpha_a = k_a^\top q \approx \epsilon_a \mu_a^\top q \approx \epsilon M$$

While the value of the third equation oscillates based on the ϵ_a.

Now for calculating the c vector we have:

$$c = \sum_{i=1}^{n} v_i \alpha_i \Rightarrow c = \alpha_a v_a + \alpha_b v_b$$

$$\Rightarrow c = \frac{\exp(\epsilon M)}{\exp(M) + \exp(\epsilon M)} v_a + \frac{\exp(M)}{\exp(M) + \exp(\epsilon M)} v_b$$

Now we analyze what happens if ϵ oscillates between 0 and 2 using limits. M is a large positive integer, so $\exp(M) \approx +\infty$

When $\epsilon \to 0$:

$$\lim_{\epsilon \to 0} c = \frac{\exp(0)}{\exp(M) + \exp(0)} v_a + \frac{\exp(M)}{\exp(M) + \exp(0)} v_b = \frac{1}{1+\infty} v_a + \frac{\infty}{1+\infty} v_b = v_b$$

When $\epsilon \to 2$:

In this case $\exp(2M)$ becomes magnificently larger than $\exp(M)$, because it is in the second order, and it is noted by ∞^2. Now we calculate the limit:

$$\lim_{\epsilon \to 2} c = \frac{\exp(2M)}{\exp(M) + \exp(2M)} v_a + \frac{\exp(M)}{\exp(M) + \exp(2M)} v_b$$

$$= \frac{\infty^2}{\infty + \infty^2} v_a + \frac{\infty}{\infty + \infty^2} v_b$$

If we neglect the first order infinity, we have:

$$\lim_{\epsilon \to 2} c = \frac{\infty^2}{\infty^2} v_a + \frac{1}{\infty^2} v_b = v_a$$

In conclusion, the overall value of vector c can vary heavily with the value of ϵ, and based on the sampling. In this case the single-headed attention fails to be resilient to this perturbation.

d.

i.

It is stated that $c = \frac{1}{2}(c_1 + c_2)$, and the requested vector for c is $c = \frac{1}{2}(v_a + v_b)$, so we have:

$$c_1 = v_a,\ c_2 = v_b$$

From part c, we know that k_i is approximately equal to μ_i, because the variance is negligible. Now our problem is transformed to two independent sub-problems, with each one exactly similar to part a and b of this section.

$$c_1 = v_a \Rightarrow k_a^\top q_1 \ggg k_i^\top q, i \notin \{a, b\} \Rightarrow q_1 = Mk_a \xrightarrow{k_a \approx \mu_a} q_1 = M\mu_a$$

$$c_2 = v_b \Rightarrow k_b^\top q_2 \ggg k_i^\top q, i \notin \{a, b\} \Rightarrow q_2 = Mk_b \xrightarrow{k_b \approx \mu_b} q_2 = M\mu_b$$

Where M is the large positive integer, used for softmax (similar to part b).

ii.

In part c.ii we saw that with $k_a \sim \mathcal{N}(1, \frac{1}{2})$, the value of c can oscillate with respect to the value which is sampled from $\mathcal{N}(1, \frac{1}{2})$, which we called it ϵ, and could oscillate between 0 and 2.

Now with two attention heads, there are two q vectors, and the c vector would always be an average of c_1 and c_2. In this way, the impact of this oscillation (vulnerability to the perturbation) is reduced. So the variance of c is reduced and it is more resilient to perturbations.

e.

i.

By using the attention equations, we have:

$$c_2 = \alpha_{21} v_1 + \alpha_{22} v_2 + \alpha_{23} v_3$$

in this case, the vector x_2 is orthogonal with every other vector (x_1 and x_3), because of the difference in their unit vectors, which are mutually orthogonal. So the inner product between every pair (with different elements) of them is equal to 0. So the above equation is approximately reduced to $c_2 = \alpha_{22} v_2$.

$$c_2 = \alpha_{22} v_2 = \frac{\exp(x_2^\top x_2) x_2}{\exp(x_1^\top x_2) + \exp(x_2^\top x_2) + \exp(x_3^\top x_2)}$$
$$\approx \frac{\exp(\beta^2) u_a}{1 + \exp(\beta^2) + 1} \approx \frac{\exp(\beta^2) u_a}{\exp(\beta^2)} = u_a = x_2$$

it is impossible to copy u_b to c_2 by just adding u_d or u_c to x_2. if we add an u_i with $i \in \{d, c\}$ to x_2, then c_2 becomes:

$$c_2 = u_a + u_i, \ i \in \{d, c\}$$

So due to the mutual orthogonality, it is impossible to reach that value by the given additions.

ii.

In order to find V, we first check the constraints given by the question.

1. $Vx_1 = v_1 = u_b$
2. $Vx_3 = v_3 = u_b - u_c$

So, V must be in a way that it's product with x_1 result in a value of u_b. So there shouldn't be any u_d terms in V so that the u_b term in x_1 is eliminated by orthogonal inner product. Also there must be a u_b term in V so that the orthogonal inner product can result in a value in terms of $+u_b$. Because the vectors are mutually orthogonal, there is no other way that u_b is not eliminated by inner product.

For the second constraint, exactly like the first constraint, it can be inferred that V must include terms with u_b and $-u_c$.

There is no need to have a u_a term, because u_a is only used in x_2, and there are no constraints on x_2. For simplicity we assume that $v_2 = 0$.

When doing the inner product, we don't want that a term gets to a second order and hence could be equivalent to β^2, So we need that the terms in V be already in a second degree (according to their outer product) so that after the inner product, the resulting terms be in the third degree, and we can divide them by β^2.

With all the reasons above, the resulted V is:

$$V = \frac{u_b u_b - u_c u_c}{\beta^2} = \frac{u_b u_b^\top - u_c u_c^\top}{\beta^2}$$

Now we calculate v_i for $i \in \{1, 2, 3\}$

$$v_1 = Vx_1 = V(u_d + u_b) = \frac{u_b u_b^\top u_b}{\beta^2} = \frac{\beta^2 u_b}{\beta^2} = u_b$$

$$v_2 = Vx_2 = V(u_a) = \frac{0}{\beta^2} = 0$$

$$v_3 = Vx_3 = V(u_c + u_b) = \frac{u_b u_b^\top u_b - u_c u_c^\top u_c}{\beta^2} = \frac{\beta^2 u_b - \beta^2 u_c}{\beta^2} = \frac{\beta^2(u_b - u_c)}{\beta^2} = u_b - u_c$$

Now we analyze the needed c vectors.

$$c_1 = u_b - u_c = v_3 = \alpha_{11} v_1 + \alpha_{12} v_2 + \alpha_{13} v_3 \xrightarrow{v_2 = 0} c_1 = v_3 = \alpha_{11} v_1 + \alpha_{13} v_3$$

$$\Rightarrow \alpha_{11} = 0,\ \alpha_{13} = 1$$

$$\Rightarrow \exp(k_3^\top q_1) \ggg \exp(k_1^\top q_1) \quad (1)$$

$$c_2 = \alpha_{21}v_1 + \alpha_{22}v_2 + \alpha_{23}v_3 = u_b = v_1 \xrightarrow{v_2=0} c_2 = \alpha_{21}v_1 + \alpha_{23}v_3$$

$$\Rightarrow \alpha_{23} = 0,\ \alpha_{21} = 1$$

$$\Rightarrow \exp(k_1^\top q_2) \ggg \exp(k_3^\top q_2) \quad (2)$$

For simplicity, I assume that $K = I$ (The identity Matrix), so we have $k_i = x_i,\ i \in \{1, 2, 3\}$.

Now for calculating Q, we use the same approach as we did for calculating the matrix V. We try to analyze and satisfy the two constraints.

For constraint (2), we have:

$$\exp(k_1^\top q_2) \ggg \exp(k_3^\top q_2) \xrightarrow{K=I} \exp(x_1^\top q_2) \ggg \exp(x_3^\top q_2)$$

$$\xrightarrow{q_i=Qx_i} \exp(x_1^\top Q x_2) \ggg \exp(x_3^\top Q x_2)$$

Now, we can see that the left side of the inequality must be bigger, and x_2 is only created from u_a. In order that the left side becomes bigger, we can make sure that the left side results in a non-zero positive value, and the right side becomes zero. So we can see that Q must have terms that result in a u_d term after calculating $Qx_2 = Qu_a$, so that the multiplication of x_1 and x_3 with this u_d term can result in the necessary inequality (left side positive, right side zero)

If we follow this procedure for constraint (1), we will reach some other constraints, that when we can use their intersection and propose a value for Q:

$$Q = \frac{u_d u_a^\top + u_c u_d^\top}{\beta^2}$$

Now we check the two constraints with the proposed Q:

$$(1):\ \exp(x_3^\top Q x_1) \ggg \exp(x_1^\top Q x_1)$$

$$\Rightarrow (u_c^\top + u_b^\top)(\frac{u_d u_a^\top + u_c u_d^\top}{\beta^2})(u_d^\top + u_b^\top) \ggg (u_d^\top + u_b^\top)(\frac{u_d u_a^\top + u_c u_d^\top}{\beta^2})(u_d^\top + u_b^\top)$$

$$\Rightarrow (u_c^\top + u_b^\top)(u_c) \ggg (u_d^\top + u_b^\top)(u_c)$$

$$\Rightarrow u_c^\top u_c \ggg 0 \Rightarrow \beta^2 \ggg 0\ \checkmark$$

$$(2):\ \exp(x_1^\top Q x_2) \ggg \exp(x_3^\top Q x_2)$$

$$\Rightarrow (u_d^\top + u_b^\top)(\frac{u_d u_a^\top + u_c u_d^\top}{\beta^2})(u_a) \ggg (u_c^\top + u_b^\top)(\frac{u_d u_a^\top + u_c u_d^\top}{\beta^2})(u_a)$$

$$\Rightarrow (u_d^\top + u_b^\top)(u_d) \ggg (u_c^\top + u_b^\top)(u_d)$$
$$\Rightarrow u_d^\top u_d \ggg 0 \Rightarrow \beta^2 \ggg 0 \checkmark$$

So the provided value for Q is correct and satisfies both inequalities.

The final answers are as follows.

$$V = \frac{u_b u_b^\top - u_c u_c^\top}{\beta^2}$$

$$K = I$$

$$Q = \frac{u_d u_a^\top + u_c u_d^\top}{\beta^2}$$

2 Pretrained Transformer models and knowledge access

d.

- **Dev Set:** 6 out of 500, 1.2%
- **"London" Baseline:** 25 out of 500, 5%

f.

- **Dev Set:** 111 out of 500, 22.2%

g.

i.

- **Dev Set:** 119 out of 500, 23.7$\bar{9}$%

ii.

This self-attention mechanism doesn't include any token-to-token interactions, and is implemented in a single (or sometimes 2) layer[1]. So it performs poorly on long distance relations between words, compared to key-query-value self-attention.

[1] Synthesizer: Rethinking Self-Attention in Transformer Models

3 Considerations in pretrained knowledge

a.

The pretrained model used a lot more data, and could perform a much better generalization due to the transfer learning procedure. The data that the model was pretrained on also contains relevant information about the birthplace, but in a much more volume, this helps the model to learn a lot of basic features of the corpus on the pretraining data, and then learns some specific features from the target dataset (which is usually small in size).

b.

When a place is just made up by the model, it can pass on false information about a place that doesn't exist in real life, and link a person's birthplace to that made up place. Also, with the model making up places, there is a strong risk of bias towards people with different characteristics, somehow similar to the bias problem we discussed during the word2vec algorithm.

c.

It is possible that when the model sees an unseen name, it relates this name to the closest name which is already seen (closeness in their embedding vectors and latent space maybe?), and generates the birthplace according to that closest seen name in the corpus. The concern can be similar to part b of this section, the model producing false information, because in reality birthplace doesn't have anything to do with similarity between peoples' names.