

## سوال ① Sentiment analysis

چون می‌تواند جملات طولانی باشد و برای درک احساس جمله نیاز به کسر ارتباطات تله‌ها با فاصله فرازدهم نیز داریم.

چون LSTM درای پارامترهای learnable است، پس می‌تواند dependency های بین‌ترينی استخراج کند و آنرا از حافظه محدودی مبتلا نداشته باشیم، لغزشی‌تری بسته به GRU است.

2) استفاده از مدل‌های LSTM بدلیل اینکه پارامترهای قابل یادگیری بیشتر باعث شدن غرایند یادگیری و نیاز بیشتر به GPU می‌شود. پس به جای اینکه بهتر است از مدل‌های GRU به پارامترهای کمتری (اردو همچنین مخلوط) معقولی در درک dependency کلمات را نیز استخراج می‌کند.

$$\begin{array}{c} \text{برای RAM سطح} \\ \hline \text{LSTM مقدار} \end{array} \quad \bar{y}^{(t)}, a^{(t)}, c^{(t)} \quad (1)$$

update gate:

$$\begin{aligned} \gamma_u^{(t)} &= W_u \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} + b_u & \left\{ \begin{array}{l} W_u = [W_{ua} \quad W_{ux}] \\ x^{(t)} = [x_0 \quad x_1 \quad \dots \quad x_n] \end{array} \right. \\ \Gamma_u^{(t)} &= \sigma(\gamma_u^{(t)}) \end{aligned}$$

forget gate:

$$\gamma_f^{(t)} = w_f \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} + b_f$$

$$\Gamma_f^{(t)} = \sigma(\gamma_f^{(t)})$$

Candidate value:

$$P\tilde{C}^{(t)} = w_c \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} + b_c$$

$$\tilde{C}^{(t)} = \tanh(P\tilde{C}^{(t)})$$

memory cell value:

$$C^{(t)} = \Gamma_u^{(t)} \tilde{C}^{(t)} + \Gamma_f^{(t)} C^{(t-1)}$$

Output gate:

$$\gamma_o^{(t)} = w_o \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix} + b_o$$

$$\Gamma_o^{(t)} = \sigma(\gamma_o^{(t)})$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial c^{(t)}} = d_{c_{\text{heat}}} + \frac{\partial L}{\partial a^{(t)}} \times \frac{\partial a^{(t)}}{\partial c^{(t)}} \\ \frac{\partial a^{(t)}}{\partial c^{(t)}} = I_0^{(t)} * (1 - \tanh^2 c^{(t)}) \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial \tilde{c}^{(t)}} = \frac{\partial L}{\partial c^{(t)}} \times \frac{\partial c^{(t)}}{\partial \tilde{c}^{(t)}} \\ \frac{\partial c^{(t)}}{\partial \tilde{c}^{(t)}} = I_u^{(t)} \\ \frac{\partial L}{\partial \tilde{c}^{(t)}} = \left[ d_{c_{\text{heat}}} + \frac{\partial L}{\partial a^{(t)}} * I_0^{(t)} * (1 - \tanh^2 c^{(t)}) \right] * I_u^{(t)} \end{array} \right.$$

$$\frac{\partial L}{\partial p\tilde{c}^{(t)}} = \frac{\partial L}{\partial \tilde{c}^{(t)}} \times \frac{\partial \tilde{c}^{(t)}}{\partial p\tilde{c}^{(t)}} = \frac{\partial L}{\partial \tilde{c}^{(t)}} \times \underbrace{\left( 1 - \tanh^2 p\tilde{c}^{(t)} \right)}_{1 - (\tilde{c}^{(t)})^2}$$

$$\left\{ \frac{\partial L}{\partial \gamma_u^{(t)}} = \frac{\partial L}{\partial c^{(t)}} \times \frac{\partial c^{(t)}}{\partial I_u^{(t)}} \times \frac{\partial I_u^{(t)}}{\partial \gamma_u^{(t)}} \right.$$

$$\left. \frac{\partial L}{\partial \gamma_u^{(t)}} = \frac{\partial L}{\partial c^{(t)}} \times \tilde{c}^{(t)} \times \left( I_u^{(t)} \times (1 - I_u^{(t)}) \right) \right)$$

$$\left\{ \frac{\partial L}{\partial \gamma_f^{(t)}} \quad \frac{\partial L}{\partial c^{(t)}} \times \frac{\partial c^{(t)}}{\partial \Gamma_f^{(t)}} \times \frac{\partial \Gamma_f^{(t)}}{\partial \gamma_f^{(t)}} \right.$$

$$\left. \frac{\partial L}{\partial \gamma_f^{(t)}} = \frac{\partial L}{\partial c^{(t)}} \times \tilde{c}^{(t)} \times (\Gamma_f^{(t)} \times (1 - \Gamma_f^{(t)})) \right)$$

$$\left\{ \frac{\partial L}{\partial \gamma_o^{(t)}} = \frac{\partial L}{\partial a^{(t)}} \times \frac{\partial a^{(t)}}{\partial \Gamma_o^{(t)}} \times \frac{\partial \Gamma_o^{(t)}}{\partial \gamma_o^{(t)}} \right.$$

$$\left. \frac{\partial L}{\partial \gamma_o^{(t)}} = \frac{\partial L}{\partial a^{(t)}} \times \tanh c^{(t)} \times (\Gamma_o^{(t)} \times (1 - \Gamma_o^{(t)})) \right)$$

$$dC_{prev} = \frac{\partial L}{\partial c^{(t-1)}} = \frac{\partial L}{\partial c^{(t)}} \times \frac{\partial c^{(t)}}{\partial c^{(t-1)}} = \frac{\partial L}{\partial c^{(t)}} \times \Gamma_f^{(t)}$$

$$\left\{ d\alpha_{prev} = \frac{\partial L}{\partial a^{(t-1)}} = \frac{\partial L}{\partial p\tilde{c}^{(t)}} \times \frac{\partial p\tilde{c}^{(t)}}{\partial a^{(t-1)}} + \frac{\partial L}{\partial \gamma_u^{(t)}} \times \frac{\partial \gamma_u^{(t)}}{\partial a^{(t-1)}} \right.$$

$$\left. + \frac{\partial L}{\partial \gamma_f^{(t)}} \times \frac{\partial \gamma_f^{(t)}}{\partial a^{(t-1)}} + \frac{\partial L}{\partial \gamma_o^{(t)}} \times \frac{\partial \gamma_o^{(t)}}{\partial a^{(t-1)}} \right)$$

$$\left\{ d\alpha_{prev} = W_{ca}^T \frac{\partial L}{\partial p\tilde{c}^{(t)}} + W_{ua}^T \frac{\partial L}{\partial \gamma_u^{(t)}} + W_{fa}^T \frac{\partial L}{\partial \gamma_f^{(t)}} \right.$$

$$\left. + W_{oa}^T \frac{\partial L}{\partial \gamma_o^{(t)}} \right)$$

$$\frac{\partial L}{\partial a^{(t)}} = W_{ca}^T \times \frac{\partial L}{\partial p \tilde{c}^{(t)}} + W_{ua}^T \times \frac{\partial L}{\partial \gamma_u^{(t)}} + W_{fa}^T \frac{\partial L}{\partial \gamma_f^{(t)}}$$

$$+ W_{oa}^T \times \frac{\partial L}{\partial \gamma_o^{(t)}}$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial w_c} = \frac{\partial L}{\partial p \tilde{c}^{(t)}} \times \frac{\partial p \tilde{c}^{(t)}}{\partial w_c} = \frac{\partial L}{\partial p \tilde{c}^{(t)}} \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix}^T \end{array} \right.$$

$$\left. \begin{array}{l} \frac{\partial L}{\partial w_u} = \frac{\partial L}{\partial \gamma_u^{(t)}} \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix}^T \end{array} \right.$$

$$\left. \begin{array}{l} \frac{\partial L}{\partial w_f} = \frac{\partial L}{\partial \gamma_f^{(t)}} \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix}^T \end{array} \right.$$

$$\left. \begin{array}{l} \frac{\partial L}{\partial w_o} = \frac{\partial L}{\partial \gamma_o^{(t)}} \times \begin{bmatrix} a^{(t-1)} \\ x^{(t)} \end{bmatrix}^T \end{array} \right.$$

$$\left. \begin{array}{l} \frac{\partial L}{\partial w_y} = \frac{\partial L}{\partial z^{(t)}} \times a^{(t)} \end{array} \right.^T$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial b_c} = \sum \frac{\partial L}{\partial p \tilde{c}^{(t)}} , \quad \frac{\partial L}{\partial b_u} = \sum \frac{\partial L}{\partial \gamma_u^{(t)}} \end{array} \right.$$

$$\left. \begin{array}{l} \frac{\partial L}{\partial b_f} = \sum \frac{\partial L}{\partial \gamma_f^{(t)}} , \quad \frac{\partial L}{\partial b_o} = \sum \frac{\partial L}{\partial \gamma_o^{(t)}} , \quad \frac{\partial L}{\partial b_y} = \sum \frac{\partial L}{\partial z^{(t)}} \end{array} \right.$$

سؤال ②

Vocab size = 30000

Seq length = 2048

embedding token = 1024

Batch = 32

Hidden vector = 768

encoder block = 12

decoder block = 8

Head number = 4

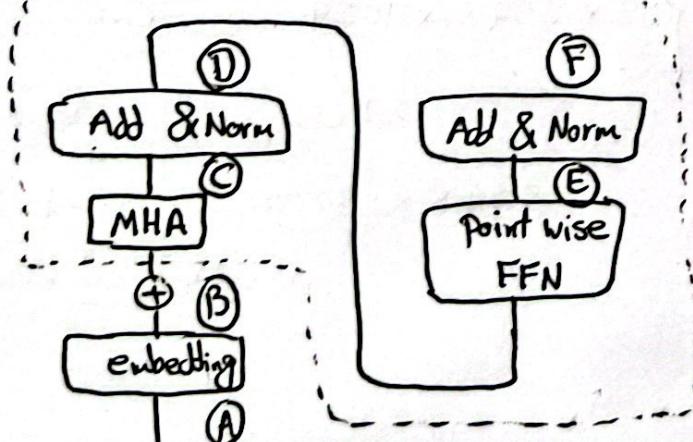
# FF layers = 2

(1) قبل از لایه embedding چون vocab size برابر 30000 است پس هر تون

سین 30000 داشته است و بعد از این لایه ب 1024 کاهش می یابد.

چون 32 بچ داریم و هر جمله مالیم از 2048 کلمه تحلیل مدد است پس ورودی اولیه  
بیلکل  $32 \times 2048 \times 30000$  بوده است.

Decoder



$$A: 32 \times 2048 \times 30000$$

$$B: 32 \times 2048 \times 1024$$

$$C: \begin{cases} 32 \times 2048 \times 192 & \text{head} \\ 32 \times 2048 \times 768 & \text{value head} \\ 32 \times 2048 \times 1024 & \text{linear} \end{cases}$$

$$D: 32 \times 2048 \times 1024$$

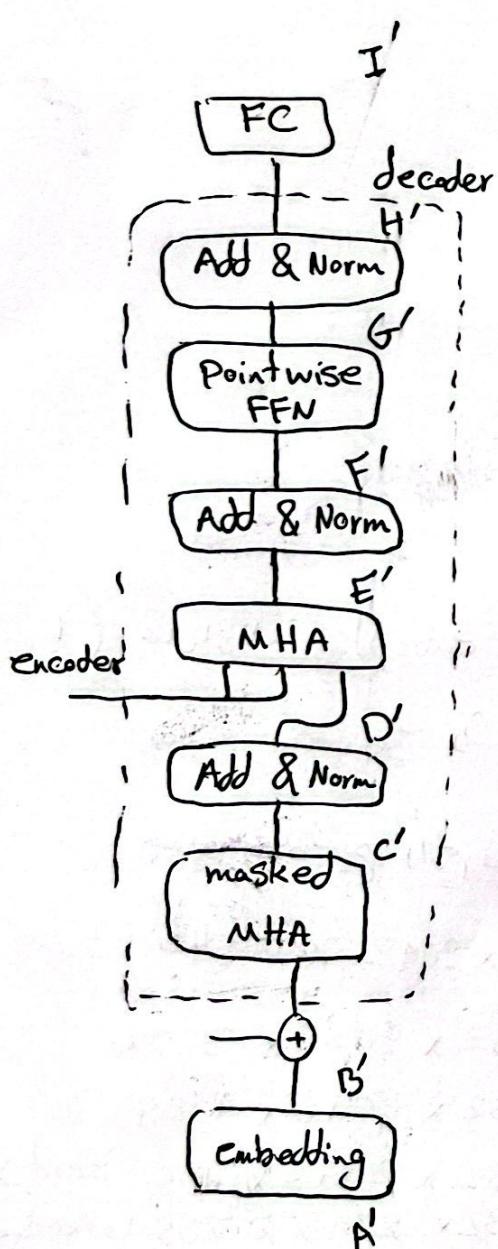
در لایه FFN ابتدا یک لایه ابعاد را از 1024 به 512 تغییر داد و در لایه دوم به 1024 بازگردانید.

$$E: \begin{cases} 32 \times 2048 \times 512 & \text{بعد از لایه اول} \\ 32 \times 2048 \times 1024 & \text{بعد از لایه دوم} \end{cases}$$

لایه Add & Norm ابعاد را تغییر می‌دهد.

$$F: \begin{cases} 32 \times 2048 \times 1024 \end{cases}$$

برای هر encoder معمورت جدا ابعاد بین محاسبه می‌شود. و تا سیزی روی هم ندارند.



$$A': 32 \times 2048 \times 30000$$

$$B': 32 \times 2048 \times 1024$$

$$C': \begin{cases} 32 \times 2048 \times 192 & \text{head ۱۶} \\ 32 \times 2048 \times 768 & \text{head ۶۴ concat} \\ 32 \times 2048 \times 1024 & \text{linear بعد از} \end{cases}$$

$$D': 32 \times 2048 \times 1024 \quad \text{خط}^{\circ}$$

$$E': \begin{cases} 32 \times 2048 \times 192 \\ 32 \times 2048 \times 768 \\ 32 \times 2048 \times 1024 & \text{linear بعد از} \end{cases}$$

$$F': 32 \times 2048 \times 1024$$

$$G': \begin{cases} 32 \times 2048 \times 512 & \text{لایه اول} \\ 32 \times 2048 \times 1024 & \text{لایه دوم} \end{cases}$$

$$H': 32 \times 2048 \times 1024 \quad \text{خط}^{\circ}$$

بعد از مخفی در ابعاد تا سیزی ندارد.

$$I': 32 \times 2048 \times 30000 \quad \begin{array}{l} \text{لایه FC بسیار} \\ \text{تغییر می‌کند} \end{array}$$

Vocab size

transformer میں کیا ہے (2)

embedding  $(30000, 1024)$ :  $30000 \times 1024 + 1024 = 30,721,024$

encoder:  $w^V, w^K, w^Q$ :  $1024 \times 192 + 192 = 196,800$  head،  
کو

$w^O$ :  $768 \times 1024 + 1024 = 787,456$

FFN:  $(1024, 512), (512, 1024)$

$(1024 \times 512 + 512) + (512 \times 1024 + 1024) =$

= 1,050,112

~VQ2 Add & Norm:  $2 \times 1024 = 2048$

مجموع 2 دفعہ  $2 \times 2048 = 4096$

کل پلاس ۱۲ ہے جس کے بعد encoder

head کو

$w^V \times 3 \times 4 + w^O + FFN + 2 (\text{Add & Norm})$

=  $3 \times 4 \times 196,800 + 787,456 + 1,050,112 + 4096$

= 4,203,264

$12 \times 4,203,264$

کل ۱۲ اندر، سطح پر ایسا برابر است،

= 50,439,168

$$2 \times \left( W^V \times 3 \times \frac{4}{4} + W^O \right) + \text{FFN} + 3(\text{Add \& Norm})$$

: decoder گوئیل ۰ تعداد

$$2 \left( 3 \times 4 \times 196,800 + 787,456 \right) + 1,050,112 + 3 \times 2048 \\ = 7,354,368$$

$$8 \times 7,354,368 = 58,834,944 : \text{decoder} \in 8 \text{ گوئیل} \text{ تعداد}$$

$$\underline{109,274,112} : \text{decoder و encoder} \text{ جمع}$$

$$\text{FC: } 1024 \times 30,000 + 30,000 = 30,750,000$$

$$170,716,160 : \text{embedding بیرون FC و دو لایه} \text{ جمع پارامترها}$$

$$201,466,160 : \text{embedding درون FC و دو لایه} \text{ جمع پارامترها}$$

$$y_n = \sum_{m=1}^N a_{nm} x_m, \quad a_{nm} = \frac{e^{x_n^T x_m}}{\sum_{m=1}^N e^{x_n^T x_m}} \quad (1)$$

$$\forall n \neq m : x_n^T x_m = 0$$

$$a_{nm} = \begin{cases} \frac{e^{x_n^T x_m}}{N-1 + e^{x_n^T x_n}} & n=m \\ \frac{1}{N-1 + e^{x_n^T x_n}} & n \neq m \end{cases} \quad \text{لذلک: } a_{nn} = \frac{e^{x_n^T x_n}}{N-1 + e^{x_n^T x_n}}$$

لذلک:  $a_{nn} = \frac{e^{x_n^T x_n}}{N-1 + e^{x_n^T x_n}}$

$$a_{nn} = \begin{cases} 1 & n=m \\ 0 & n \neq m \end{cases} \rightarrow a_{nn} = 1$$

$$y_n = \sum_{m=1}^N a_{nm} x_m = \underbrace{a_{nn} x_n}_1 = x_n \quad \text{لذلک: } a_{nn} = 1$$

$$\text{Var}(X) = E(X^2) - E^2(X) \quad (2)$$

چون  $a$  و  $b$  هردو از توزیع نرمال ب میانگین صفر و واریانس ۱ آمده است.  
پس ضربه سان نیز نیز توزیع نرمال ب میانگین صفر و واریانس ۱ است.

$$E(a) = E(b) = E(ab) = 0$$

$$\text{Var}(ab) = \text{Var}(a) = \text{Var}(b) = 1$$

$$\text{Var}(a^T b) = \sum_i^D \text{Var}(a_i b_i) = D \times 1 = D$$

$$\begin{aligned} \text{Var}(a^T b) &= E[(a^T b)^2] - \underbrace{\mathbb{E}^2[a^T b]}_0 = \mathbb{E}[(a^T b)^2] \\ \Rightarrow \mathbb{E}[(a^T b)^2] &= D \end{aligned}$$


---

$$Y(X) = \text{Concat}[H_1, H_2, \dots, H_H] W^{(0)} \quad (3)$$

$$H_h = \text{Softmax}\left[\frac{Q_h K_h^T}{\sqrt{D_{K_h}}}\right] V_h$$

$$Q_h = X W_h^{(q)}, \quad K_h = X W_h^{(k)}, \quad V_h = X W_h^{(v)}$$

$$Y(X) = \sum_{h=1}^H \text{Softmax}\left[\frac{Q_h K_h^T}{\sqrt{D_{K_h}}}\right] X W^{(h)}$$

$$Y(X) = \sum_{h=1}^H \text{Softmax}\left[\frac{Q_h K_h^T}{\sqrt{D_{K_h}}}\right] X W_h^{(v)} W_h^{(0)}$$

$$Y(X) = \sum_{h=1}^H \text{Softmax}\left[\frac{Q_h K_h^T}{\sqrt{D_{K_h}}}\right] V_h W_h^{(0)}$$

$$Y(X) = \sum_{h=1}^H H_h W_h^{(0)} = \text{Concat}[H_1, H_2, \dots, H_H] W^{(0)}$$

## positional embedding (۱)

از کاظمی absolute vs relative

در حالت absolute، هر تکن ورودی دارای یک representation خاص برای موقعیت یا ایندکس در توالی ورودی است. چون در این حالت توالی های ورودی در زمان تنین محدود به قوی خاصی است پس در زمان ست مسئله در ارزیابی توالی با عوامل دیده نشده (بزیر لتر از ماسیم در train) دارد. قوی توالی فلیس است.

در حالت relative، فاصله نسبی بین دو تکن ورودی حساب می شود و راهی برای رابطه positional تکن ها است. اگری از محدودیت های این حالت پیشیده مدل کردن بودن پیدا شده سازی است. آنچه این حالت برای توالی با قوی سیستم نزد توانایی تخمیم دارد.

از کاظمی تابت یا learnable بودن

تابت: روئی است که تابت بوده و در قوی فرایند training تغییری نمی کند.  
مانند encoding سینوس و با توابع Sin یا Cos انجام می شود.  
چون قابل آموزش سنت ممکن است در همه زمینه ها عملکرد خوبی نداشته باشد.  
از نظر تئوری محدود به قوی خاصی سنت و چون از توابع متناوب استفاده می کند  
می توانند هر عوامل را سایپر ت کنند.

حالت learnable : این نوع embedding دارای پارامترهایی است که در قوی فرآیند training می‌توانند آموزش داده شود. بین دلیل توانی فیت سدن بهتر روی سکل های خاص را نسبت دارد و این مورد (کنی از مزیت های آن نسبت به حالت fixed است.

از محدودیت ها و مکاسب آن می‌توان به اعمال overfitting اشاره کرد و این چون پارامتر سینتیکی به شبکه آنها می‌کند پس از تحریک می‌سایع سکل تراست. با اهمی آن بدلیل اهمیت کردن حساسیت در لایه self attention بدلیل اهمیت سدن ماتریس موقتی به ماتریس Query- Key است.

## 2) روش ROPE

این روش به جای اضافه کردن بیانگر موقتی، یک چرخش برای بیانگر اعمال می‌کند مثلاً بیانگر در موقعیت ۱ با  $\theta_1$  و بیانگر در موقعیت ۲ با  $\theta_2$  می‌چرخد و موقتی سبی دو بیانگر حفظ می‌شود.

چنین محدودیت در این روش بحیره می‌شود عبارتند از :

- این اجزه را می‌توان مدل ها درودی با طول های مختلف را پردازش نمود و این ROPE انجطاف نسبت به قوی برای مدیریت اسناد طولانی کاربردی است.

- تفہیم می‌کند که واسیلی بین تولن  $\theta$  با افزایش فاصله سبی رابطه عکس داشته و ناهض می‌باشد. این میزگیری برای انتظار گرفتن داستان تولن های دور از هم در زبانهای ملعن بسیار مهم است. همچنین این ویژگی به مدل کمک می‌کند تا به واسطه های تزریک محل توجه سینتیکی نمود و همچنین واسیلی های رور هنروری را نیز داشته باشد.

- ROPE بخلاف روش های قبلی که ممکن است با مکاتریم خطي self-attention سازگار نباشد، طوری طراحی شده که بکار رفته با self-attention خطی کار کند. این مکاتریم عملکرد کلی مدل سریپردازی دالهای Sequential را بهبودی دهد.

سوال ④

$$\text{ارادتقدر بلندیم: } \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \text{ الگوریتم حرضش } (3)$$

$$f(\alpha_m) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \alpha_1^m \\ \alpha_2^m \end{bmatrix} = \begin{bmatrix} \alpha_1 \cos\theta - \alpha_2 \sin\theta \\ \alpha_1 \sin\theta + \alpha_2 \cos\theta \end{bmatrix}$$

$$\left| \alpha_m \right|_2 = \sqrt{(\alpha_1)^2 + (\alpha_2)^2} \Rightarrow \left| \alpha_m \right|_2^2 = \alpha_1^2 + \alpha_2^2$$

$$\begin{aligned} \left| f(\alpha_m) \right|_2^2 &= (\alpha_1 \cos\theta - \alpha_2 \sin\theta)^2 + (\alpha_1 \sin\theta + \alpha_2 \cos\theta)^2 \\ &= \cancel{\alpha_1^2 \cos^2\theta + \alpha_2^2 \sin^2\theta} - 2\alpha_1 \alpha_2 \cos\theta \sin\theta \\ &\quad + \cancel{\alpha_1^2 \sin^2\theta + \alpha_2^2 \cos^2\theta} + 2\alpha_1 \alpha_2 \sin\theta \cos\theta \\ &= \cancel{\alpha_1^2} (\sin^2\theta + \cos^2\theta) + \cancel{\alpha_2^2} (\sin^2\theta + \cos^2\theta) = \alpha_1^2 + \alpha_2^2 \end{aligned}$$

پس این حرضش اندیشه Vector را حفظ کن و فقط حرضش با اندازه  $\theta$  در آن فاصله می‌افتد.

این مفهوم سهان می‌باشد و RoPE Embedding magnitude بردار مقدار را تعبیر نمی‌کند و پارامتر  $\theta$  جدیدی به آن اختصاص نمی‌کند و حول آن بردارها را می‌چرخاند. این پارامتر  $\theta$  و حرضش بردار نسبت نسبی (relative) بوده و الگوریتم همراه با هم بسیاری به یک میزان دیگر حرضش می‌سوند. و اگر باهم نباشند، حرضش باعث فاصله نسبی آن‌ها از هم خواهد شد.

سوال ④

۵) روش Alibi به این صورت است که باید  $q$  به حاصلضرب  $K^T$  اضافه می‌کند.

باید  $q$  که اضافه می‌کند بصورت  $[0 \ 1 \ 2 \ \dots \ m-1]$  می‌باشد.

بردار  $[0 \ 1 \ 2 \ \dots \ m-1]$  نشان (لطفاً فاصله سبی تولن) ام از بعضی تولن هاست به این صورت که نزدیکترین تولن به تونکن ام فاصله ۰ و همینطور  $\frac{فاصله}{(1-n)} \cdot (ارد)$ .

ضریب  $m$  نیز به منظور Scale کردن این فوایل سبی استفاده می‌شود و بصورت تجربی از ابتدا انتخاب می‌شود و محدودیت  $1 < m < 0$  در نظر گرفته می‌شود.

سوال ⑤

$$Y = \text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_K}}\right)V \quad (1)$$

در ساخت Self Attention، ماتریس‌های  $Q$ ،  $V$  و  $K$  ماتریس‌های با اندازه  $n \times d$  دارند. که هر دوام بین لغتی (نیازهای از تولن) می‌باشند. هر سطر آن یک توکن را بیان می‌کند. می‌توانیم قوی نیازی Attention را بصورت fully connected در نظر بگیریم.

به این صورت که عروقی شبکه ماتریس  $V$  باشد و وزن‌های خروجی‌های شبکه با استفاده از  $\text{Softmax}\left(\frac{QK^T}{\sqrt{D_K}}\right)$  بسته آید که یک ماتریس  $n \times n$  است و خروجی سیزمان توجه روی تولن‌های مهم را با خروجی بزرگتر نشان می‌دهد.

ماتریسی  $Q, V, K$  با اندازه  $N \times D$  هستند. (2)

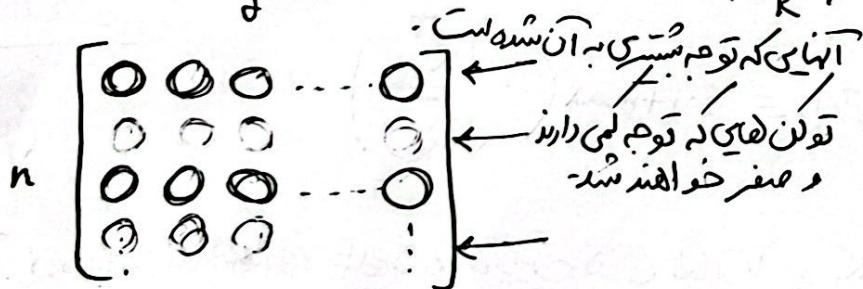
$Q$  یک ماتریس  $N \times N$  خواهد بود و خروجی $\backslash$ مل  $N \times D$  خواهد بود.

چون اردر پارامتری  $O(n^2)$  است پس تعداد پارامترهای

برابر  $N^2 D^2$  است. پس از اردر  $(N^2 D^2)$  خواهد بود.

(3) به ازای هر تکن ورودی برای شبکه self-attention می‌توانم یک لایه در نظر بگیریم که حاوی یک Query و Key است. بحثت وجود Softmax آن توجه شود که بسیار کم جل بودن به صفر تبدیل خواهد شد و ماتریس Sparse خواهد بود.

$$\text{Attention}(Q, K, V) = \underset{d}{\text{Softmax}} \left( \frac{QK^T}{\sqrt{D_K}} \right) V \quad (N, D) \text{ shape}$$



(4) (ریتمی) Self-attention تکن های ورودی مجزا و مستقل از هم در نظر گرفته می‌شوند و آرپوزیشنال ایندکس (positional encoding) ندانسته باشند، ورودی را بصورت تکنی در نظر نمی‌گیرند. درنتیجه آرچای تکن های (H) را عوض کنیم، ترتیب خروجی آن ها نیز به همین صورت تغییر می‌کند و این موضعی متریوم equivariant بودن این مدل‌گردی را می‌رساند. این مدل‌گردی به معنی این توجه کل تکن را در نظر نمایی و ترتیب براسن اهمیت ندارد. وقتی همسایه‌ای بالغه‌های پرکار در جملات خروجی آن را محاسبه می‌کند.