دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری ماشین
پاییز ۱۴۰۳
استاد: علی شریفی زارچی

- پایانترم -

۱. (۱۵ + ۱۰ نمره) در هر بخش، درستی یا نادرستی گزینه را مشخص کرده و کامل توضیح دهید.

الف) در هنگام آموزش هرگونه مدل روی هر نوع مساله، از زمانی که تعداد پارامترها متناسب با تعداد ورودی‌ها شود به حداقل خطای تست می‌رسیم و پس از آن با افزایش تعداد پارامترها همیشه شاهد روند افزایشی خطای تست خواهیم بود.

ب) معماری RNN سریع‌تر از Transformer است ولی دقت آن نسبت به Transformer کم‌تر است.

ج) ابعاد فضای Embedding متن و تصویر در CLIP می‌تواند متفاوت باشد.

د) در یادگیری انتقالی با CNN، زمانی که توزیع داده‌های هدف مشابه توزیع داده‌های پیش‌آموزش باشد، معمولاً چند لایه‌ی Fully Connected آخر جایگزین شده و دوباره آموزش داده می‌شود، در حالی که وزن لایه‌های قبلی ثابت می‌مانند.

ه) اگر یک مدل شبکه‌ی عصبی ۳ لایه برای رسیدن به دقت ۹۰ درصد روی یک مساله‌ی ورودی نیاز به ۱ میلیارد پارامتر داشته باشد، یک شبکه‌ی عصبی ۱۰ لایه برای رسیدن به همان سطح دقت روی همان مساله نیاز به تقریبا همان تعداد پارامتر دارد که البته عرض هر لایه کاهش می‌یابد.

و) اگر از یک میلیارد تصویر داده شده، فقط یک میلیون تصویر برچسب داشته باشد برای خوشه‌بندی (Clustering) از کل یک میلیارد تصویر می‌توان استفاده کرد ولی برای دسته‌بندی (Classification) صرفا می‌توانیم از یک میلیون تصویر برچسب‌دار استفاده کنیم.

ز) برای یک CNN با ابعاد کرنل ۵ × ۵ و استراید ۱ و روی تصاویر ۱۰۰۰ × ۱۰۰۰ داشتن سه لایه برای استخراج ویژگی‌های سراسری از تصویر کافی است.

ح) استفاده از Patch ها به صورت انکد شده به عنوان ورودی ترانسفورمر ViT کافی نیست و حتما باید از Positional Embedding استفاده کنیم.

حل.

- الف) نادرست. با افزایش تعداد پارامترها، ممکن است ابتدا مدل دچار بیش‌برازش (Overfitting) شود، اما خطای تست همیشه روند افزایشی ندارد و بستگی به تنظیمات مدل و داده‌ها دارد. مدل‌هایی مثل Dropout یا Regularization می‌توانند به کاهش خطای تست کمک کنند.
- ب) نادرست. معماری Transformer به دلیل طراحی موازی و استفاده از Self-Attention معمولاً سریع‌تر از RNN است، به‌ویژه در پردازش متن طولانی، اگرچه از نظر دقت نیز برتری دارد.
- ج) درست. در مدل CLIP، ابعاد فضای Embedded متن و تصویر می‌توانند متفاوت باشند و مدل با یک فضای مشترک آن‌ها را نگاشت می‌کند.
- د) درست. در یادگیری انتقالی با CNN، معمولاً لایه‌های Fully Connected آخر جایگزین شده و لایه‌های اولیه به‌دلیل یادگیری ویژگی‌های عمومی ثابت می‌مانند.
- ه) نادرست. افزایش عمق (تعداد لایه‌ها) معمولاً نیاز به تعداد پارامترهای کمتری دارد، مگر اینکه تکنیک‌هایی مثل کاهش عرض لایه یا بهینه‌سازی ساختاری انجام شود.

- و) درست. Clustering می‌تواند از کل داده‌ها استفاده کند زیرا نیازی به برچسب ندارد، اما Classification نیازمند داده‌های برچسب‌دار است.
- ز) نادرست. سه لایه برای استخراج ویژگی‌های سراسری از تصاویر با ابعاد ۱۰۰۰×۱۰۰۰ کافی نیست؛ لایه‌های بیشتری لازم است تا بتوان اطلاعات سطح بالا را استخراج کرد.
- ح) درست. در ViT، استفاده از Positional Embedding ضروری است زیرا Transformer به خودی خود ترتیب مکانی پچ‌ها را تشخیص نمی‌دهد.

۲. (۱۵ نمره)

الف) ما به یک مسئله رگرسیون خطی تک‌بعدی خاص علاقه‌مند هستیم. داده‌های مربوط به این مسئله شامل n نمونه $(x_۱, y_۱), \ldots, (x_n, y_n)$ هستند که x_i و y_i اعداد حقیقی برای همه i می‌باشند. فرض کنید $\mathbf{w}^* = [w_۰^*, w_۱^*]^T$ جواب حداقل مربعات باشد که به دنبال آن هستیم. به عبارت دیگر، $\mathbf{w}^*$ کمینه‌کننده عبارت زیر است:

$$J(\mathbf{w}) = \frac{۱}{n}\sum_{i=۱}^{n}(y_i - w_۰ - w_۱x_i)^۲.$$

می‌توانید فرض کنید که برای مقاصد ما، جواب یکتا است.
هر کدام از گزاره‌های زیر که اگر $\mathbf{w}^* = [w_۰^*, w_۱^*]^T$ پاسخ کمترین مربعات باشد، درست است مشخص کنید و کامل توضیح دهید.

$$\frac{۱}{n}\sum_{i=۱}^{n}(y_i - w_۰^* - w_۱^*x_i)y_i = ۰$$

$$\frac{۱}{n}\sum_{i=۱}^{n}(y_i - w_۰^* - w_۱^*x_i)(y_i - \bar{y}) = ۰$$

$$\frac{۱}{n}\sum_{i=۱}^{n}(y_i - w_۰^* - w_۱^*x_i)(x_i - \bar{x}) = ۰$$

$$\frac{۱}{n}\sum_{i=۱}^{n}(y_i - w_۰^* - w_۱^*x_i)(w_۰^* + w_۱^*x_i) = ۰$$

که در آن $\bar{x}$ و $\bar{y}$ میانگین نمونه‌ها بر اساس مجموعه داده هستند.

ب) چندین آماره از داده‌ها وجود دارند که می‌توانیم برای تخمین $\mathbf{w}^*$ استفاده کنیم. این اعداد عبارتند از:

$$\bar{x} = \frac{۱}{n}\sum_{i=۱}^{n}x_i$$

$$\bar{y} = \frac{۱}{n}\sum_{i=۱}^{n}y_i$$

$$C_{xx} = \frac{۱}{n}\sum_{i=۱}^{n}(x_i - \bar{x})^۲$$

$$C_{xy} = \frac{۱}{n}\sum_{i=۱}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$C_{yy} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

فرض کنید فقط به مقدار w_1^* اهمیت می‌دهیم. می‌خواهیم w_1^* را بر اساس تنها دو آماره از آماره‌های بالا تعیین کنیم. کدام دو آماره برای این کار لازم است؟ توضیح دهید.

حل.

الف) مشتق گرفتن از $J(\mathbf{w})$ نسبت به w_1 و w_0 شرایط بهینگی زیر را می‌دهد:

$$\frac{\partial}{\partial w_0}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i) = 0$$

$$\frac{\partial}{\partial w_1}J(\mathbf{w}) = \frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i = 0$$

این به این معناست که خطای پیش‌بینی $(y_i - w_0 - w_1 x_i)$ با هیچ تابع خطی از ورودی‌ها همبسته نیست (میانگین صفر دارد و با ورودی‌ها تغییر نمی‌کند). $(x_i - \bar{x})$ و $(w_0^* + w_1^* x_i)$ هر دو توابع خطی از ورودی‌ها هستند.

ب) به C_{xx} (پراکندگی x) و C_{xy} (وابستگی خطی بین x و y) نیاز داریم. نیازی به توجیه نبود زیرا این پایه‌های اساسی در درس ظاهر شده‌اند. اگر بخواهیم این را بیشتر از نظر ریاضیاتی استخراج کنیم، می‌توانیم به یکی از پاسخ‌های سوال قبلی نگاه کنیم:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0,$$

که می‌توان آن را به صورت زیر بازنویسی کرد:

$$\left[\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x})\right] - w_0^*\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})\right] - w_1^*\left[\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x})\right] = 0.$$

با استفاده از این حقیقت که:

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) = 0$$

می‌بینیم که:

$$\frac{1}{n}\sum_{i=1}^{n} y_i(x_i - \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = C_{xy},$$

و

$$\frac{1}{n}\sum_{i=1}^{n} x_i(x_i - \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = C_{xx}.$$

جایگذاری این مقادیر در معادله بالا منجر به عبارت زیر می‌شود:

$$C_{xy} - w_1^* C_{xx} = 0.$$

۳. (۱۵ نمره)

ماتریس داده زیر را در نظر بگیرید که چهار نمونه $X_i \in R^2$ را نشان می‌دهد:

$$X = \begin{pmatrix} ۴ & ۱ \\ ۲ & ۳ \\ ۵ & ۴ \\ ۱ & ۰ \end{pmatrix}$$

الف) جهت‌های مولفه اصلی با طول واحد X را محاسبه کنید و بیان کنید که اگر فقط به دنبال یک مولفه اصلی باشیم، الگوریتم PCA کدام یک را انتخاب می‌کند.

ب) بهترین (با حداقل خطای بازسازی) projection از X را در یک زیرفضای یک بعدی با مبدا صفر پیدا کنید.

حل.

جواب اول:

الف)

$$x' = \frac{1}{4}\begin{pmatrix} 4+2+5+1 \\ 1+3+4+0 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} 12 \\ 8 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$\Rightarrow \tilde{X} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 2 & 2 \\ -2 & -2 \end{pmatrix}, \quad \tilde{X}^T = \begin{pmatrix} 1 & -1 & 2 & -2 \\ -1 & 1 & 2 & -2 \end{pmatrix} \Rightarrow \tilde{X}^T\tilde{X} = \begin{pmatrix} 10 & 6 \\ 6 & 10 \end{pmatrix}$$

$$\times \frac{1}{4} \downarrow$$

$$\begin{pmatrix} 2.5 & 1.5 \\ 1.5 & 2.5 \end{pmatrix}$$

$$\Rightarrow \det\begin{pmatrix} 2.5-\lambda & 1.5 \\ 1.5 & 2.5-\lambda \end{pmatrix} = 0 \rightarrow \boxed{\lambda_1 = 1 \text{ و } \lambda_2 = 4}$$

$$\lambda_1 = 1 \Rightarrow \begin{pmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{pmatrix} v_1 = 0 \Rightarrow v_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} \end{pmatrix}$$

$$\lambda_2 = 4 \Rightarrow \begin{pmatrix} -1.5 & 1.5 \\ 1.5 & -1.5 \end{pmatrix} v_2 = 0 \Rightarrow v_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

از آنجایی که $\lambda_2 > \lambda_1$، PCA مولفه v_2 را انتخاب می‌کند.

ب)

$$X' = XA = \begin{pmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 5 \\ 5 \\ 9 \\ 1 \end{pmatrix}$$

جواب دوم:

الف)

$$\Sigma = \frac{1}{n-1} \tilde{X}^T \tilde{X} = \frac{1}{n-1} \begin{pmatrix} 10 & 6 \\ 6 & 10 \end{pmatrix} = \frac{2}{3} \begin{pmatrix} 5 & 3 \\ 3 & 5 \end{pmatrix}$$

$$\Rightarrow \det \begin{pmatrix} 10/3 - \lambda & 2 \\ 2 & 10/3 - \lambda \end{pmatrix} = 0 \Rightarrow \boxed{\lambda_1 = \frac{16}{3},\ \lambda_2 = \frac{4}{3}}$$

$$\lambda_1 = \frac{16}{3} \Rightarrow v_1 = \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} \leftarrow \text{انتخاب PCA}$$

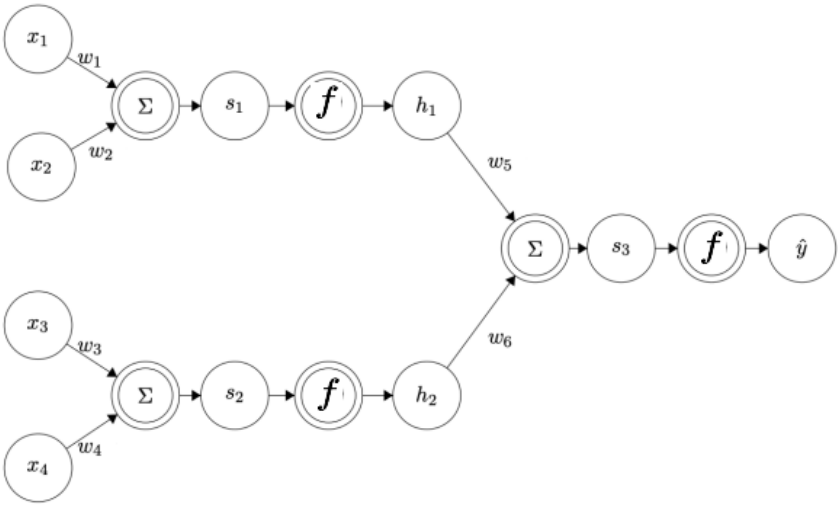
$$\lambda_2 = \frac{4}{3} \Rightarrow v_2 = \begin{pmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{pmatrix}$$

ب)

$$X' = XA = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 2 & 2 \\ -2 & -2 \end{pmatrix} \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2\sqrt{2} \\ -2\sqrt{2} \end{pmatrix}$$

۴. (۲۰ نمره)

شبکه عصبی زیر را در نظر بگیرید. در این شبکه، گره‌های تک‌دایره‌ای نشان‌دهنده متغیرها (به عنوان مثال x_1 متغیر ورودی، h_1 متغیر میانی و $\hat{y}$ متغیر خروجی است.) و گره‌های دودایره‌ای نشان‌دهنده توابع هستند.



فرض کنید تابع هزینه L_2 به صورت $L(y, \hat{y}) = ||y - \hat{y}||_2^2$ تعریف شده است. اگر یک داده با مقدار $(x_1, x_2, x_3, x_4) = (1, 2, -1, 0)$ و برچسب واقعی ۱ داشته باشیم، از الگوریتم *backpropagation* برای محاسبه مشتق جزئی L نسبت به w_1 استفاده کنید.

$$f(x) = ReLU(x) - 2ReLU(x-1) + 0.5ReLU(x+1)$$
$$(w_1, w_2, w_3, w_4, w_5, w_6) = (1, -0.25, 0, 1, 2, -1)$$

حل. ابتدا مقادیر متغیرها را در حرکت رو به جلو محاسبه می‌کنیم:

$$s_1 = 0.5$$
$$s_2 = 0$$
$$h_1 = 1.25$$
$$h_2 = 0.5$$
$$s_3 = 2$$
$$\hat{y} = 1.5$$

سپس برای الگوریتم پس انتشار (*backpropagation*) داریم:

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial s_3} \times \frac{\partial s_3}{\partial h_1} \times \frac{\partial h_1}{\partial s_1} \times \frac{\partial s_1}{\partial w_1}$$
$$= 2||\hat{y} - y|| \times f'(s_3) \times w_5 \times f'(s_1) \times x_1$$
$$2(1.5 - 1) \times (-0.5) \times 2 \times 1.5 \times 1 = -1.5$$

۵. (۱۵ نمره)

الف) ابعاد تصویر ورودی را برابر (n_h, n_w, n_c) درنظر بگیرید. بر روی این ورودی یک لایه کانولوشن با فیلتر ۱×۱ با استراید و پدینگ استفاده می‌کنیم. کدام یک از عبارات زیر صحیح است؟ (تمام موارد صحیح را انتخاب کنید و کامل توضیح دهید.)

۱. شما می‌توانید n_c را با استفاده از کانولوشن ۱×۱ کاهش دهید. با این حال، نمی‌توانید n_h, n_w را تغییر دهید.
۲. شما می‌توانید از ماکس پولینگ استاندارد فقط برای کاهش n_h, n_w و نه n_c استفاده کنید.
۳. شما می‌توانید از کانولوشن ۱×۱ برای کاهش n_h, n_w, n_c استفاده کنید.
۴. شما می‌توانید از ماکس پولینگ برای کاهش n_c استفاده کنید.

ب) فرض کنید یک فیلتر ۲×۲ را با $stride = ۲$ روی یک تصویر ۴×۴ اعمال می‌کنیم. این عملیات کانولوشن را با ضرب یک ماتریس در بردار به صورت $Ax = z$ خلاصه کنید که در اینجا: x همان تصویر اولیه و z همان نقشه ویژگی پس از اعمال فیلتر خواهد بود.

a۱۱	a۱۲	a۱۳	a۱۴
a۲۱	a۲۲	a۲۳	a۲۴
a۳۱	a۳۲	a۳۳	a۳۴
a۴۱	a۴۲	a۴۳	a۴۴

$\times$

w۱	w۲
w۳	w۴

حل.

الف) گزینه ۱ و ۲

ب) از آنجایی که با توجه به صورت سوال x بردار و A ماتریس است، بنابراین x' نیز که حاصل ضرب بردار و ماتریس است، بردار می‌شود. ابتدا تصویر اولیه را به یک بردار تبدیل می‌کنیم (این تبدیل به هر صورتی و با هر ترتیبی صحیح است). با توجه به بردار x بدست آمده برای اینکه حاصل ضرب Ax همان تاثیر فیلتر دو در دو را داشته باشد، کافی است ماتریس A را به صورت زیر تعریف کنیم:

$$X = \begin{bmatrix} a_{۱,۱} \\ a_{۱,۲} \\ a_{۱,۳} \\ a_{۱,۴} \\ a_{۲,۱} \\ a_{۲,۲} \\ a_{۲,۳} \\ a_{۲,۴} \\ a_{۳,۱} \\ a_{۳,۲} \\ a_{۳,۳} \\ a_{۳,۴} \\ a_{۴,۱} \\ a_{۴,۲} \\ a_{۴,۳} \\ a_{۴,۴} \end{bmatrix}$$

$$A = \begin{bmatrix} w_1 & w_3 & 0 & 0 & w_2 & w_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_1 & w_3 & 0 & 0 & w_2 & w_4 & 0 & 0 \\ 0 & 0 & w_1 & w_3 & 0 & 0 & w_2 & w_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_1 & w_3 & 0 & 0 & w_2 & w_4 \end{bmatrix}$$

$$x' = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

۶. (۲۰ نمره)

مساله embed کردن متن را در نظر بگیرید. فرض کنید تعداد واژگان دیکشنری برابر ۱۰۰۰۰ باشد و تعداد توکن‌های متن ورودی حداکثر ۵۰۰ در نظر گرفته شده باشد و واژه‌ها به صورت one-hot به عنوان ورودی به شبکه داده شوند.

اگر از معماری ترنسفورمر استفاده شود، فرض کنید که بُعد embedding هر لایه self-attention برابر ۷۶۸، بعد مخفی یا همان تعداد گره‌های لایه مخفی MLP تک لایه مخفی برابر ۳۰۰۰ و تعداد head لایه self-attention برابر ۴ و بعد متناظر با هر head برابر ۱۹۲ باشد. فرض کنید از positional embedding با استفاده از توابع سینوسی و کسینوسی ایجاد شده است. شمای کلی شبکه‌ی embedding و یک بلوک انکدر را رسم کنید و سپس تعداد پارامترهای آن را به دست آورید.

حل. مقدمه و توضیحات کلی

- تعداد واژگان (Vocab) = ۱۰۰۰۰
- حداکثر طول ورودی (Sequence Length) = ۵۰۰
- ابعاد تعبیه (Embedding Dimension) = ۷۶۸
- تعداد نورون‌های لایه‌ی پنهان در مسیر MLP (یا FFN) = ۳۰۰۰
- تعداد هد در Self-Attention = ۴ (بعد هر هد = ۱۹۲)
- تعبیه‌ی جایگاه (Positional Embedding) از نوع سینوسی/کسینوسی (بدون پارامتر)

ورودی به‌صورت one-hot با ابعاد $[\text{Batch}, \text{SeqLen}, ۱۰۰۰۰]$ در نظر گرفته می‌شود. لایه‌ی امبدینگ واژگان (Token Embedding) در عمل همان ضرب یک ماتریس $\mathbf{W}_{\text{embed}}$ با بردار ورودی one-hot است. خروجی بخش تعبیه (شامل جمع با Positional Embedding) در نهایت دنباله‌ای با ابعاد $[\text{SeqLen}, ۷۶۸]$ خواهد بود.

توضیح برای تصحیح‌کننده:

- **در تصحیح، بخش اصلی مربوط به بلاک Attention است و بعد از آن FFN اهمیت بیشتری دارد.**
- **نسبت به Residualها و لایه‌نرمالیزیشن (LayerNorm) می‌توان با دست بازتر برخورد کرد و حساسیت کمتری داشت.**

۱) پارامترهای لایه‌ی Embedding

- **ماتریس توکن امبدینگ** $\mathbf{W}_{\text{embed}}$ با ابعاد $[۱۰۰۰۰ \times ۷۶۸]$:

$$\#\text{params} = ۱۰۰۰۰ \times ۷۶۸ = ۷,۶۸۰,۰۰۰.$$

- بایاس معمولاً برای لایه‌ی امبدینگ در اکثر پیاده‌سازی‌ها وجود ندارد یا بسیار نادر است. پس $\#\text{params}_{\text{bias}} = ۰$.
- **تعبیه‌ی جایگاه** (Positional Embedding) از نوع سینوسی است؛ درنتیجه پارامتری برای یادگیری ندارد.

کل پارامترهای بخش Embedding $\approx$ ۷,۶۸۰,۰۰۰

۲) پارامترهای Multi-Head Self-Attention

با توجه به بعد ۷۶۸ در ورودیِ لایه، برای هر یک از سه ماتریس $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$:

$$\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{۷۶۸\times ۷۶۸}.$$

- **وزن**: هر یک $۷۶۸ \times ۷۶۸ = ۵۸۹,۸۲۴$.
- **بایاس**: هر یک ۷۶۸.

بنابراین تعداد پارامترهای هر یک از $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$:

$$۵۸۹,۸۲۴ + ۷۶۸ = ۵۹۰,۵۹۲.$$

برای سه ماتریس:

$$۳ \times ۵۹۰,۵۹۲ = ۱,۷۷۱,۷۷۶.$$

- **ماتریس خروجی (Out Projection):** $\mathbf{W}_O \in \mathbb{R}^{۷۶۸\times ۷۶۸}$ به‌علاوه‌ی بایاس سایز ۷۶۸:

$$۵۸۹,۸۲۴ + ۷۶۸ = ۵۹۰,۵۹۲.$$

جمع پارامترهای Self-Attention:

$$\underbrace{۱,۷۷۱,۷۷۶}_{Q,K,V} + \underbrace{۵۹۰,۵۹۲}_{W_O} = ۲,۳۶۲,۳۶۸.$$

۳) پارامترهای Feed-Forward Network (FFN)

بعد از ماژول Attention، مسیر FFN دو لایه‌ی خطی دارد:

$$(۷۶۸ \to ۳۰۰۰ \to ۷۶۸).$$

- $\mathbf{W}_۱ \in \mathbb{R}^{۷۶۸\times ۳۰۰۰}$, $b_۱ \in \mathbb{R}^{۳۰۰۰}$:
  وزن: $۷۶۸ \times ۳۰۰۰ = ۲,۳۰۴,۰۰۰$
  بایاس: ۳۰۰۰
  جمع: $۲,۳۰۴,۰۰۰ + ۳۰۰۰ = ۲,۳۰۷,۰۰۰$.
- $\mathbf{W}_۲ \in \mathbb{R}^{۳۰۰۰\times ۷۶۸}$, $b_۲ \in \mathbb{R}^{۷۶۸}$:
  وزن: $۳۰۰۰ \times ۷۶۸ = ۲,۳۰۴,۰۰۰$
  بایاس: ۷۶۸
  جمع: $۲,۳۰۴,۰۰۰ + ۷۶۸ = ۲,۳۰۴,۷۶۸$.

مجموع پارامترهای FFN:

$$۲,۳۰۷,۰۰۰ + ۲,۳۰۴,۷۶۸ = ۴,۶۱۱,۷۶۸.$$

۴) پارامترهای LayerNormها

در یک بلاک انکودر استاندارد، دو LayerNorm داریم:

- یکی بعد از Self-Attention (قبل یا بعد از جمع Residual)
- یکی بعد از FFN (قبل یا بعد از جمع Residual)

هر LayerNorm دو بردار پارامتر دارد (γ, β) به طول ۷۶۸:

$$\gamma \in \mathbb{R}^{۷۶۸}, \quad \beta \in \mathbb{R}^{۷۶۸}.$$

پس برای هر LayerNorm: $۷۶۸ + ۷۶۸ = ۱۵۳۶$ پارامتر.
دو لایه LayerNorm در بلاک:

$$۲ \times ۱۵۳۶ = ۳۰۷۲.$$

از آن‌جا که اهمیت کمی در تصحیح نهایی دارند، این بخش با دست باز تصحیح می‌شود.

۵) جمع‌بندی کل پارامترها

$$\begin{aligned}
\text{Token Embedding} &: ۷,۶۸۰,۰۰۰ \\
\text{Self-Attention} &: ۲,۳۶۲,۳۶۸ \\
\text{FFN (MLP)} &: ۴,۶۱۱,۷۶۸ \\
\text{LayerNorms} &: ۳,۰۷۲
\end{aligned}$$

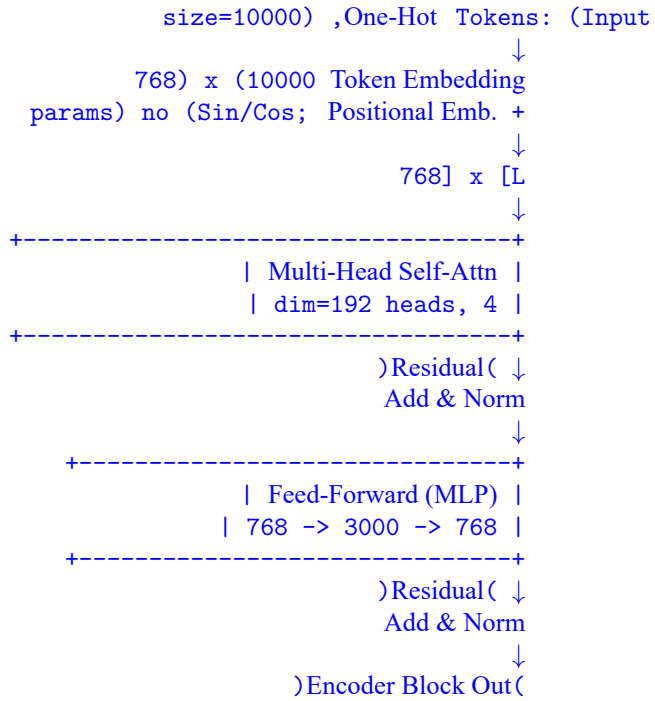
$$\text{Total} = ۷,۶۸۰,۰۰۰ + ۲,۳۶۲,۳۶۸ + ۴,۶۱۱,۷۶۸ + ۳,۰۷۲ = ۱۴,۶۵۷,۲۰۸.$$

در مجموع حدود $\boxed{\text{میلیون پارامتر}۱۴/۶۶}$ خواهیم داشت (با در نظر گرفتن وزن و بایاس در همه‌ی بخش‌ها).

۶) نمای کلی معماری

یک بلاک انکودر به‌همراه لایه‌ی امبدینگ در شکل زیر قابل نمایش است (به‌صورت خلاصه):

شمای کلی لایه‌ی امبدینگ + بلاک انکودر ترنسفورمر



پایان