Subject: Date:

علی آشتیانی ۴۰۲۶۱۵۴ یادگیری عمیق تمرین سری سوم بخش تئوری

مسئله ۱:

$$\frac{\partial E_i}{\partial w} = \frac{\partial E_i}{\partial \hat{y}_i} \times \frac{\partial \hat{y}_i}{\partial s_i} \times \frac{\partial s_i}{\partial w} = \frac{\partial E_i}{\partial \hat{y}_i} \times V \times \frac{\partial s_i}{\partial w}$$

$$\frac{\partial s_i}{\partial w} = \frac{\partial s_i}{\partial w}\Big|_{s_{i-1}=\text{ثابت}} + \frac{\partial s_i}{\partial s_{i-1}} \times \frac{\partial s_i}{\partial w} \longrightarrow \frac{\partial s_i}{\partial w} = \sum_{j=1}^{i} \frac{\partial s_i}{\partial s_j} \times \frac{\partial s_j}{\partial w}\Big|_{s_{j-1}=\text{ثابت}}$$

$$\frac{\partial s_i}{\partial s_j} = \prod_{k=j+1}^{i} \frac{\partial s_k}{\partial s_{k-1}} = \prod_{k=j+1}^{i} w = w^{i-j}$$

$$\longrightarrow \frac{\partial E_i}{\partial w} = \frac{\partial E_i}{\partial \hat{y}_i} \times V \times \left(\sum_{j=1}^{i} w^{i-j} \times \frac{\partial s_j}{\partial w}\Big|_{s_{j-1}=\text{ثابت}} \right) \longrightarrow$$

$$\frac{\partial E_r}{\partial w} = \frac{\partial E_r}{\partial \hat{y}_r} \times V \times \left(\frac{\partial s_r}{\partial w}\Big|_{s_r=\text{ثابت}} + w \times \frac{\partial s_r}{\partial w}\Big|_{s_1=\text{ثابت}} + w^2 \frac{\partial s_1}{\partial w}\Big|_{s_0=\text{ثابت}} \right)$$

$$\frac{\partial E_i}{\partial V} = \frac{\partial E_i}{\partial \hat{y}_i} \times \frac{\partial \hat{y}_i}{\partial V} \longrightarrow \frac{\partial E_r}{\partial V} = \frac{\partial E_r}{\partial \hat{y}_r} \times \frac{\partial \hat{y}_r}{\partial V}$$

$$\frac{\partial E_i}{\partial \hat{y}_i} = 1 \times k \,,\ \frac{\partial \hat{y}_i}{\partial s_i} = k \times d \,,\ \frac{\partial s_i}{\partial w} = d \times d \times d \longrightarrow \frac{\partial E_i}{\partial w} = 1 \times d \times d$$

$$\frac{\partial E_i}{\partial \hat{y}_i} = 1 \times k \,,\ \frac{\partial \hat{y}_i}{\partial V} = k \times d \times k \longrightarrow \frac{\partial E_i}{\partial V} = 1 \times d \times k$$

مسئله ۲: بله، اگر تابع attention عبارت است از یک mapping از یک query به یک مجموعه از key، value، key، query و value ها همگی بردارهایی هستند ضروری نیست که طول بردارهای value با key و query یکسان باشد. در آن صورت هر value توسط یک تابع نمایی بین key و query مربوط به آن value تعیین می‌شود. به عنوان مثال اگر طول بردارهای query، key، dk، طول بردارهای value، dv، تعداد key‌ها و value‌ها n باشد داریم:

$$\text{attention}(q, K, V) \,;\ q = 1 \times dk \,,\ K = n \times dk \,,\ V = n \times dv$$

یک علت scale کردن ضرب داخلی این است که برای مقادیر بزرگ dk، اندازه ضرب‌های داخلی بزرگ می‌شوند و در نتیجه تابع softmax به ناحیه‌هایی هدایت می‌کند که دارای شیب‌های بسیار کوچک است. scale کردن ضرب‌های داخلی با ضریب $\frac{1}{\sqrt{dk}}$ یک متغیر با این موضوع استفاده می‌شود. علت انتخاب ضریب $\frac{1}{\sqrt{dk}}$ این است که ضرب داخلی بردارهای q و k به طول dk که مولفه‌های آن‌ها متغیرهای تصادفی d بعدی با واریانس ضرب واریانس یک هستند یک متغیر تصادفی با واریانس ضرب واریانس dk است.

ج) به صورت کلی از attention چند استفاده می‌شود:

۱) در لایه‌های encoder-decoder attention، query‌ها از لایه قبلی decoder، key‌ها و value‌ها از خروجی encoder می‌آیند.

۲) encoder شامل لایه‌های self-attention است. در یک لایه self-attention، key‌ها، value‌ها و query‌ها همگی از یک جا می‌آیند که همان خروجی لایه قبلی encoder است.

Subject: Date:

۱۳) decoder نیز شامل لایه‌های self-attention است که این بار query‌ها از خروجی لایه قبل decoder می‌آیند. تفاوت این است که در این بار یک مقابله‌ی auto-regressive بین از میان اطلاعات سمت چپ decoder جلوگیری می‌شود.

۱۴) با استفاده از ترتیب دنباله‌ها، اطلاعات مکانی token‌ها به این صورت به دنباله تزریق شده است که در ابتدای بلوک‌های encoder و decoder، پس از ایجاد embedding‌ها از input‌ها، positional embedding‌های به صورت یک تفاوت بالا در position به embedding‌ها افزوده می‌شود.

انتخاب جالب یک positional embedding‌ها وجود دارد در این مقاله از توابع سینوسی و کسینوسی با فرکانس‌های مختلف استفاده شده است.

$$PE(pos, 2i) = \sin(pos / 10000^{2i/d_model})$$

$$PE(pos, 2i+1) = \cos(pos / 10000^{2i/d_model})$$

در این مقاله همچنین از positional embedding‌های یادگرفته شده نیز استفاده شده است که نتایج مشابهی داده است.

۱۵) در این مقاله سه معیار مقایسه گرفته شده است:

۱) پیچیدگی محاسباتی هر لایه، پیچیدگی محاسباتی هر لایه self attention برابر $O(n^2 d)$ است که مشابه مدل‌های دنباله به دنباله بازگشتی d است (اگر n کوچکتر از بعد نهانی d است) اگر مشابه‌ی است از RNN $O(n \cdot d^2)$، CNN $O(k \cdot n \cdot d^2)$ است.

۲) تعداد عملیات موازی شدنی، در self-attention، CNN، همه عملیات‌های تولید به صورت موازی انجام شدنی یعنی تعداد عملیات موازی‌ای برابر $O(1)$ است در حالی که در RNN این تعداد برابر $O(n)$ است.

۳) بیشترین طول مسیر، بیشترین طول مسیر در self-attention برابر $O(1)$ است در RNN برابر $O(n)$ و در CNN برابر $O(\log_k(n))$ است.

۱۶) معماری transformer از دو بخش encoder و decoder تشکیل شده است هر بخش از stack شدن $N = 6$ بلوک یکسان ایجاد می‌شود. در بخش encoder، این بلوک از دو لایه‌ی self-attention و یک شبکه FC تشکیل می‌آید. بلوک بخش decoder شبیه بلوک بخش encoder است با این تفاوت که دو لایه attention چند سر پیاده شده. برای آموزش مدل از بهینه‌ساز Adam با نرخ آموزش متغیر و تکنیک‌های regularization، residual dropout، label smoothing استفاده شده است.