حمید رضا امیرزاده
۴۰۱۲۰۶۹۹۹

موضوع: تمارین سری چهارم درس دیپ لرنینگ

① اگر ماتریس های $W_1 \in \mathbb{R}^{K \times m}$ و $W_2 \in \mathbb{R}^{m \times K}$ با رنک کامل SVD تجزیه کنیم داریم:

$$W_1 = U_1 \Sigma_1 V_1^T \quad , \quad W_2 = U_2 \Sigma_2 V_2^T$$

تابع هزینه به صورت زیر است:

$$L(W_1, W_2 : X) = \frac{1}{n} \|X - W_2 W_1 X\|_F^2 + \lambda \|W_1\|_F^2 + \lambda \|W_2\|_F^2$$

اگر ترم های مربوط به منظم سازی وجود نداشتند، مینیمم کردن تابع هزینه منجر به این می شد که $W_2 W_1$ مانند یک ماتریس نگاشت دهنده عمل کند که X را به K بزرگترین Singular Vector های آن نگاشت کند. بنابراین می توان گفت که $W_2 W_1$ تقریباً یک ماتریس همانی است بنابراین W_1 و W_2 تقریباً معکوس هم هستند. حال می توان ترم های منظم ساز را به صورت زیر نوشت:

$$\lambda \left(\|W_1\|_F^2 + \|W_2\|_F^2 \right) = \lambda \left(\|\Sigma_1\|_F^2 + \|\Sigma_2\|_F^2 \right) = \lambda \sum_{i=1}^{K} \left(\sigma_i^2 + \frac{1}{\sigma_i^2} \right)$$

مینیمم کردن عبارت بالا به صورت $\left(\sigma^2 + \frac{1}{\sigma^2}\right)' = 0 \rightarrow -\frac{2}{\sigma^3} + 2\sigma = 0$ نتیجه می دهد که $\sigma = 1$

بنابراین می توان نتیجه گرفت که W_1 و W_2 pseudo-inverse هستند و همچنین singular value های غیر صفر آن ها نیز نزدیک به یک است. این بدان معناست که ردیف های W_1 تقریباً متعامد و همچنین ستون های W_2 نیز تقریباً متعامد هستند. در نهایت می توان گفت که با این شرایط بهترین حالت برای یک inductive bias برای W_2 داریم که این است که تقریباً ماتریس W_1 pseudo-inverse باشد.

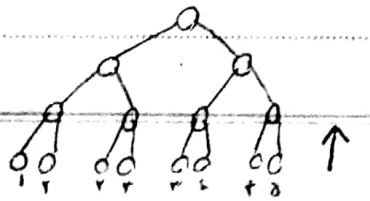
---

② الف) تابع هزینه بازسازی $J_1 = \frac{1}{2} \|x' - x\|^2$ هر دو بردار ورودی x_1 و x_2 را در نظر می گیرد و از اینکه شبکه به کدام یک از آن ها توجه نکند جلوگیری می کند. بنابراین اگر تمام بردارهای ورودی را در نظر بگیرد برای بهبود بردار p.

ب) $W_1 \in \mathbb{R}^{D_p \times 2D_x}$, $b_1 \in \mathbb{R}^{D_p \times 1}$, $W_2 \in \mathbb{R}^{2D_x \times D_p}$, $b_2 \in \mathbb{R}^{2D_x \times 1}$

$W_3 \in \mathbb{R}^{D_c \times D_p}$, $b_3 \in \mathbb{R}^{D_c \times 1}$ تعداد پارامترها $= 4 D_p D_x + D_p + 2 D_x + D_c D_p + D_c$

ج) می توان جمله ورودی را به چند کلمات مجاور تقسیم کرد. پس بردار p هر یک از این جمع ها را به دست آورد. پس به همین ترتیب بازگشتی همین کار را روی بردارهای p به دست آمده انجام می دهیم. این کار را ادامه می دهیم تا به یک بردار p واحد برسیم. در واقع به صورت درختی متقابل کار می کنیم.



تاریخ: ..... موضوع: .....

۲) الف) اگر ابعاد برداری کدیک باشد آنگاه توانایی decoder برای مدل کردن در روش‌های general و tanh layer یکسان است. اما در ابعاد بزرگتر tanh layer توانایی مدل کردن بهتری دارد.
tanh layer و general از نظر پیچیدگی محاسباتی تقریبا با هم برابرند. در روش dot با پارامتر اضافه‌ای وجود ندارد. اما در general تعداد پارامترهای W_a برابر $|d| \times |e|$ است. در روش tanh layer تعداد پارامترهای V_a^T برابر D_{W_a} و تعداد پارامترهای W_a برابر $D_{W_a} \times D_{d+e}$ است.
از لحاظ هزینه محاسباتی dot نسبت به دو روش دیگر کمتر است اما توانایی مدل کردن کمتری دارد.
همچنین در مرحله Backprop نیز هزینه کمتری دارد. در روش tanh layer خروجی score را tanh اعمال شده از حالت خطی خارج می‌کند و خروجی در بازه $[-1, 1]$ قرار می‌دهد. بنابراین می‌تواند از explode شدن گرادیان جلوگیری کند و همچنین مشکل Vanish شدن گرادیان را حل کند.
در شبکه‌های seq2seq مدل general از دو روش دیگر بهتر است چون نسبت به dot توانایی مدل کردن بیشتری دارد و همچنین نسبت به tanh layer هزینه محاسباتی کمتری دارد.

ب) مکانیزم اول سیر عمیق و پرهزینه‌ای دارد که مطابق زیر است:

$$h_d^{t-1} \rightarrow \alpha^t \rightarrow c^t \rightarrow h_d^t$$

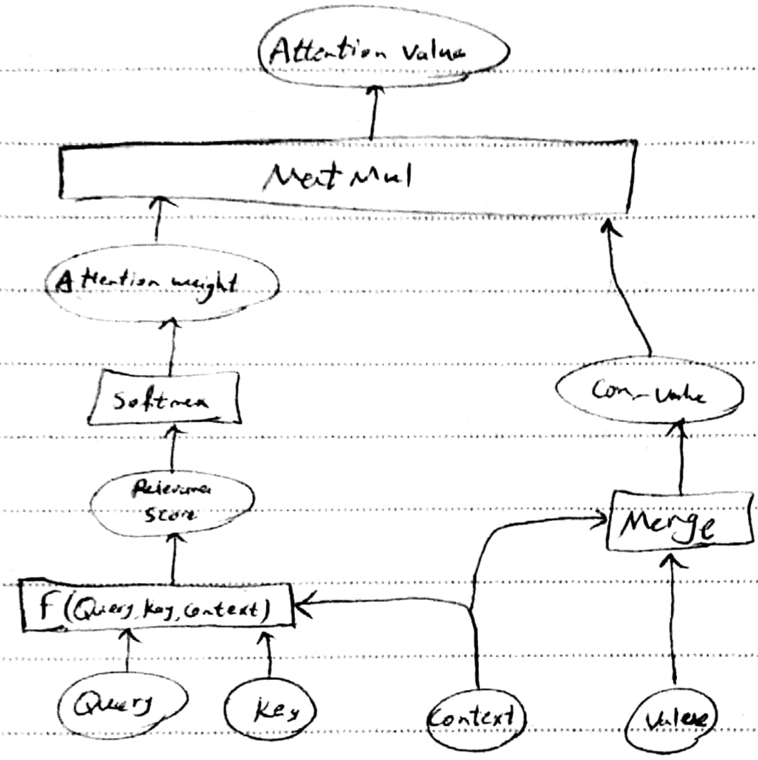
اما مکانیزم دوم سیر کم‌هزینه‌تری دارد که به صورت زیر است:

$$h_d^t \rightarrow \alpha^t \rightarrow c^t \rightarrow \tilde{h}_d^t$$

مکانیزم اول در encoder از روش bidirectional استفاده شده است که باعث دید توانایی در دادن بازگشتی می‌شود، اما در مکانیزم دوم از روش یک طرفه استفاده شده است.
در مکانیزم اول برای محاسبه score از روش concat استفاده شده اما در مکانیزم دوم از چندین روش از جمله dot، general، concat، location استفاده شده که location نتیجه بهتری می‌دهد.
به‌طور کلی می‌توان گفت مکانیزم دوم نسبت به روش اول برتری دارد.

ج) روش‌های مبتنی بر توجه برای محاسبه میزان توجه صرفاً به tokenهایی اعمال که هست تمرکز می‌کنند.

راه حلی که برای این مسئله ارائه شده است روش tri-attention نام دارد. در این روش به جای محاسبه شباهت بین [redacted] Query و Key مقدار شباهت سه‌گانه برای محاسبه شباهت بین Query و Context و Key می‌سازد که امتیاز بالاتری context کل جمله را در نظر نیز در محاسبه شباهت دخیل می‌کند. همچنین Value نیز با Context و merge شده و یک contextual value داریم. در نهایت مانند روش‌های قبل از شباهت Context و Query و Key و مقدار Value جدید برای محاسبه توجه استفاده می‌کنیم.



الف)

$N \rightarrow$ sequence length

$D \rightarrow$ dim. of Q and K

$M \rightarrow$ dim. of V

① اگر از استاندارد softmax self attention استفاده کنیم آنگاه داریم:

Computation Cost : $O(N \max(D, M))$

Memory : $O(NM + ND) + O(N^2) + O(N^2 M) + O(NM) = O(N^2 M + ND)$

↓ K, Q, V ذخیره — ↓ ذخیره $Q_i^T K_j$ — ↓ ذخیره $\exp(\frac{Q_i^T K_j}{\sqrt{D}}) V_j$ — ↓ ذخیره V_j

ب)

$$sim(q, K) = (q^T K + 1)^2$$

ج)

$$\phi(x) = [1, x_1, x_2, \dots, x_D, x_1^2, x_2^2, \dots, x_D^2, \sqrt{2}x_1x_2, \dots \sqrt{2}x_{D-1}x_D]^T$$

د)

$$V_i' = \frac{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)}{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)} V_j$$

هزینه محاسباتی برای یک کرنل چندجمله‌ای درجه ۲ برابر $O(ND^2M)$ است، همچنین حافظه مورد نیاز برابر $O(NM \max(D, D))$ است، می‌توان گفت که اگر $N > D^2$ آنگاه استفاده از این رابطه بهتر از رابطه قبلی می‌باشد.