سوال اول)ثابت کنید فرمول تابع هزینه ی naive-softmax همانند تابع هزینه ی cross-entropy بین y و $\hat{y}$ است.

ابتدا باید متغیر های نظیر هم در دو رابطه را بشناسیم:

y بردار خروجی ای است که باید به آن برسیم و در آن درایه ی مربوط به کلمه ی مرکزی ۱ و برای بقیه ی کلمات ۰ هستند. در واقع نمایش *one-hot* برای کلمه ی مرکزی است.

$\hat{y}$ بردار خروجی مدل است که هر درایه ی آن شامل احتمال *context* بودن یکی از کلمات واژه های موجود را برای کلمه ی مرکزی است.

$\hat{y}_o$ درایه ی مربوط به کلمه ی مرکزی در بردار $\hat{y}$ است که برابر با احتمال *P(o|c)* است.

این را هم میدانیم که $\log(1) = 0$.

از فرمول cross-entropy شروع می‌کنیم و سپس عبارت مربوط به کلمه ی عبارت مرکزی را از بقیه ی عبارت ها که صفر میشوند جدا میکنیم و با جایگذاری مقادیر معادل y_o به فرمول naive-softmax میرسیم:

$$J_{cross-entropy} = -\sum_{w \in Vocabulary} y_w \log(\hat{y}_w) = -y_o \log(\hat{y}_o) - \sum_{\substack{w \in Vocabulary, \\ w\,!= o}} y_w \log(\hat{y}_w)$$

$$= -\log(\hat{y}_o) = -\log(P(O = o|C = c)) = J_{naive-softmax}$$

سوال دوم)(i)محاسبه ی مشتق جزئی $J_{cross-entropy}(v_c, o, U)$ نسبت به v_c:

$$\frac{\partial J(v_c, o, U)}{\partial v_c} = \frac{\partial}{\partial v_c}\left(-\log(P(o|c))\right) = -\frac{\partial}{\partial v_c} \log \frac{\exp(u_o^T \cdot v_c)}{\sum_{w \in V} \exp(u_w^T \cdot v_c)}$$

$$= -\frac{\partial}{\partial v_c}\left(\log(\exp(u_o^T \cdot v_c)) - \log\left(\sum_{w \in V} \exp(u_w^T \cdot v_c)\right)\right)$$

* log تابع معکوس exp است.

$$= \overset{\mathrm{I}}{\frac{-\partial}{\partial v_c} \log(\exp(u_o^T v_c))} + \overset{\mathrm{II}}{\frac{\partial}{\partial v_c} \log\left(\sum_{w \in V} \exp(u_w^T \cdot v_c)\right)} = ?$$

$$(\mathrm{I}) = \frac{-\partial}{\partial v_c} u_o^T \cdot v_c = -u_o^T \rightarrow -u_o$$

chaine rule

$$(\mathrm{II}) = \frac{\partial \log\left(\sum_{w \in V} \exp(u_w^T \cdot v_c)\right)}{\partial \sum_{w \in V} \exp(u_w^T \cdot v_c)} \cdot \frac{\partial \sum_{w \in V} \exp(u_w^T \cdot v_c)}{\partial v_c}$$

$$= \frac{1}{\sum_{x \in V} \exp(u_x^T \cdot v_c)} \cdot \sum_{w \in V} \exp(u_w^T \cdot v_c) . u_w$$

ایندکس ها باید عوض بشه تا باهم ساده نشوند.

$$(?) \Rightarrow\ ? = (\mathrm{I}) - (\mathrm{II}) = -u_o + \frac{\sum_{w \in V} \exp(u_w^T \cdot v_c)\, u_w}{\sum_{x \in V} \exp(u_x^T \cdot v_c)}$$

$y = Softmax(u_w^T \cdot v_c)$

$$= -u_o + \sum_{w \in V} u_w \frac{\exp(u_w^T \cdot v_c)}{\sum_{x \in V} \exp(u_x^T \cdot v_c)} = -u_o + \sum_{w \in V} u_w \cdot \overset{\hat{y}}{P(w|c)}$$

ضرب داخلی بردار U در y

$$= -u_o + U . \hat{y}^T$$

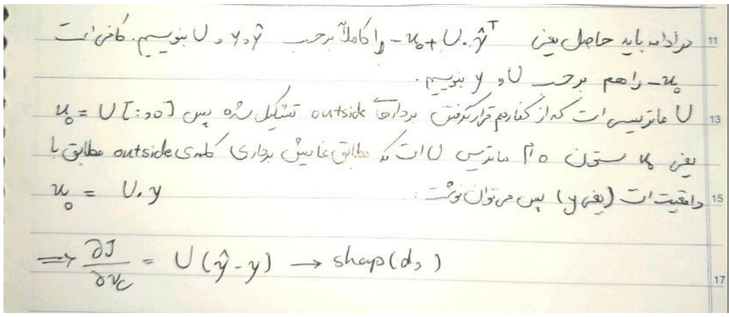
بررسی بعدهای هر ماتریس :

$U \rightarrow d \times |V|$ $\quad u_o \rightarrow d \times 1$

$\hat{y}^T \rightarrow |V| \times 1$ $\quad$ حاصل ضرب $\rightarrow d \times 1$

$v_c \rightarrow d \times 1$

d به اندازه ی یک ستون از ماتریس U است.



در ادامه باید حاصل یعنی $-u_o + U.\hat{y}^T$ را کاملاً برحسب $\hat{y}$ و y و U بنویسیم. کافی است
u_o را هم برحسب U و y بنویسیم.
U ماتریسی است که از کنار هم قرار گرفتن بردارهای outside تشکیل شده پس $u_o = U[:, o]$
یعنی u_o ستون oام ماتریس U است که مطابق نمایش برداری کلمه‌ی outside مطابق با
واقعیت است (یعنی y) پس می‌توان نوشت: $u_o = U.y$

$$\Rightarrow \frac{\partial J}{\partial v_c} = U(\hat{y}-y) \rightarrow shap(d,)$$

سوال دوم)(ii)حاصل سوال قبل چه زمانی صفر میشود؟

برای پیدا کردن مقادیر vc که در آن گرادیان برابر با صفر است، ما باید معادله زیر را حل کنیم:

U * (y^ - y) = 0

این معادله دارای دو حالت است: زمانی که U غیرقابل معکوس کردن باشد، که این اتفاق می‌تواند در صورتی رخ دهد که ستون‌های U با یکدیگر رابطه خطی داشته باشند. به عبارت دیگر، اگر دو یا بیشتر از دو ستون U با هم رابطه خطی داشته باشند، آنگاه ماتریس U مقدار کامل نیست و معادله U * (y^ - y) = 0 بی‌نهایت جواب خواهد داشت.

در این مورد، مشتق جزئی Jnaive-softmax نسبت به vc می تواند به ازای مقادیر متعدد vc، که با نقاط بهینه ی محلی متفاوت تابع مطابقت دارد، صفر باشد. با این حال، اگر U دارای ستون‌های مستقل خطی باشد، معادله تنها راه‌حل y^ - y = 0 را خواهد داشت، به این معنی که مشتق فقط زمانی صفر خواهد بود که y^ = y. به این معنی vc در حال حاضر در حد مطلوب است و خروجی مدل با خروجی واقعی برابر شده و نیازی به آپدیت بیشتر نیست.

سوال دوم)(iii)نحوه ی تاثیر گرادیان بدست آمده روی vc:

این مشتق تفاضل بین دو عبارت است: *U * y ^* و *U * y*

- **عبارت *U * y ^* یک مجموع وزن‌دار از بردارهای کلمات *outside* است، جایی که وزن‌ها براساس اختلاف بین احتمالات پیش‌بینی و احتمالات برچسب واقعی داده شده‌اند. هر بردار *outside* بر اساس سهم آن در پیش‌بینی واژه مرکزی *c* وزن‌گذاری می‌شود. این عبارت به وکتور *vc* کمک می‌کند که به سمت بردارهای *outside* که احتمالاً بیشترین تکرار را در متن دارند حرکت کند. به طور کلی، این عبارت وکتور *vc* را تشویق می‌کند تا بهترین پیش‌بینی را برای واژگان *outside* داشته باشد.**
- **عبارت *U * y* صرفاً یک مجموع وزنی از بردارهای بیرونی است، که در آن وزن ها با احتمالات برچسب واقعی داده می شوند. هر بردار *outside* با توجه به سهم واقعی خود در *context* کلمه مرکزی *c* وزن دهی می شود. این عبارت به عنوان *baseline* عمل می کند و نشان دهنده سهم هر کلمه *outside* در بافت کلمه مرکزی *c* است. با کم کردن این عبارت از عبارت قبلی، گرادیانی به دست می‌آوریم که *vc* را از جهت‌هایی که به *context* کلمه مرکزی کمک نمی‌کنند، دور میکند و به سمت جهت‌هایی که در یادگیری پیش‌بینی کلمات *outside*در متن موثر تر هستند، می‌کشد.**

بنابراین، هنگامی که این گرادیان از کلمه ***vector vc*** کم می شود، بردار را به گونه ای به روز می کند که توانایی آن را برای پیش بینی کلمات بیرونی در متن کلمه مرکزی بهبود می بخشد. در حالی که حساسیت آن را نسبت به جهت هایی که به زمینه مرتبط نیستند نیز کمتر می کند.

همچنین میتوان اینگونه تفسیر کرد که گرادیان بدست آمده را می توان به عنوان یک عبارت تصحیح تفسیر کرد که کلمه مرکزی بردار ***vc*** را به منظور بهبود پیش بینی مدل تنظیم می کند. عبارت اول ($\hat{y}$ - y) نشان دهنده خطا یا عدم تطابق بین مقادیر هدف پیش بینی شده و واقعی است. عبارت دوم ***U*** نشان دهنده سهم هر بردار کلمه متنی نسبت به خروجی پیش بینی شده است. با کم کردن گرادیان از بردار کلمه فعلی، می توانیم ***embedding*** را به گونه ای به روز کنیم که ***context*** کلمات اطراف را بهتر نشان دهد.

سوال دوم)(iv) در بسیاری از موارد word embedding، از بردار های نرمال شده ی L2 به جای بردار خام آنها استفاده میشود. مسئله ی رده بندی عبارت ها به صورت مثبت و منفی را درنظر بگیرید. چه زمانی روش نرمال سازی L2 اطلاعات مفید را از بین میبرد؟ چه زمانی نه؟

نرمال سازی ***L2*** معمولاً در برنامه های کاربردی ***downstream*** استفاده می شود زیرا می تواند به کاهش اثر طول های مختلف بردارهای ورودی کمک کند و مقایسه و تجزیه و تحلیل جاسازی های مختلف را آسان تر می کند. با این حال، مواردی وجود دارد که عادی سازی ***L2*** به طور بالقوه می تواند اطلاعات مفیدی را برای تسک ***downstream*** طبقه بندی عبارات به عنوان مثبت یا منفی از بین ببرد.
یکی از این موارد زمانی است که نرمال سازی ***L2*** برای ***embedding*** هایی اعمال می شود که به طور خطی مستقل نیستند، مانند سناریویی که $ux = \alpha uy$ برای کلماتی که $x \neq y$ ومقدار اسکالری مانند α . در ایین حالت، نرمال سازی ***embedding*** ها منجر به این می شود که هر دو بردار جهت یکسانی داشته باشند، عملاً دو بردار در یک بردار جمع شوند و تمایز بین آنها از بین برود. این می‌تواند منجر به از دست رفتن اطلاعات مهمی شود که برای طبقه‌بندی عبارات به عنوان مثبت یا منفی مرتبط است.
از سوی دیگر، زمانی که تعبیه‌ها به صورت خطی مستقل هستند، نرمال‌سازی ***L2*** اطلاعات مفیدی را برای تسک ***downstream*** از بین نمی‌برد. در واقع، می‌تواند با کاهش تأثیر فیچرهای نامربوط در بردارهای ورودی و تأکید بر ویژگی‌های مهم، به بهبود عملکرد طبقه‌بندی کمک کند. در چنین مواردی، نرمال‌سازی ***L2*** می‌تواند به‌ویژه هنگام برخورد با ***embedding*** های با ابعاد بالا که ممکن است مستعد ***overfit*** یا ***underfit*** باشند، مفید باشد.
اگر فاصله نسبی بین بردارهای کلمه مهم باشد، نرمال سازی ***L2*** می تواند اطلاعات مفیدی را برای کار ***downstream*** از بین ببرد. به عنوان مثال، اگر دو کلمه ***x*** و ***y*** را در نظر بگیریم که به روش خاصی به یکدیگر مرتبط هستند (مثلاً "***king***" و "***queen*** ")، بردارهای نرمال شده ***L2*** آنها ممکن است از نظر جهت شبیه تر از همتایان غیرنرمال شده ی آنها شوند. این می تواند تشخیص رابطه صحیح بین کلمات را برای تسک های ***downstream*** مانند قیاس کلمات یا تشابه دشوارتر کند.
با این حال، نرمال سازی ***L2*** می تواند در زمینه های دیگر نیز مفید باشد که بزرگی مطلق بردارهای کلمه مهم نیست. به عنوان مثال، در تحلیل احساسات، جهت بردارهای کلمه ممکن است مهمتر از بزرگی آنها باشد. در چنین مواردی،

نرمال سازی *L2* ممکن است به بهبود عملکرد کار *downstream* با کاهش تأثیر نقاط پرت یا فیچر های نامربوط کمک کند. علاوه بر این، نرمال سازی *L2* همچنین می تواند به بهبود پایداری و تعمیم مدل با کاهش تأثیر نمونه های آموزشی خاص یا نویز کمک کند.

سوال سوم) مشتقات جزئی Jnaive-softmax(vc,o,U) را با توجه به هر یک از بردارهای کلمه "outside"، uw محاسبه کنید.

$$J_{naive-softmax}(v_c, o, U) = -\log(\hat{y}_o) \quad , \quad \hat{y}_o = \frac{\exp(u_o^T \cdot v_c)}{\sum_{w \in V} \exp(u_w^T \cdot v_c)}$$

V: Vocabulary سایز

$$J = -u_o^T \cdot v_c + \log\left(\sum_{w \in V} \exp(u_w^T \cdot v_c)\right)$$

$$\frac{\partial J(v_c, o, U)}{\partial u_w} = \frac{-\partial}{\partial u_w}\left(u_o^T \cdot v_c - \log \sum_{w \in V} \exp(u_w^T \cdot v_c)\right)$$

$$= \frac{-\partial (u_o^T \cdot v_c)}{\partial u_w} + \frac{\partial\left(\log \sum_{w \in V} \exp(u_w^T \cdot v_c)\right)}{\partial u_w}$$

$$\frac{v_c \exp(u_w^T \cdot v_c)}{\sum_{i \in V} \exp(u_i^T \cdot v_c)}$$

$$= -v_c(y - \hat{y})$$

۱) $\omega \neq o$:

$$\frac{\partial J}{\partial u_w} = 0 + \frac{v_c \exp(u_w^T \cdot v_c)}{\sum_{i \in V} \exp(u_i^T \cdot v_c)} = v_c \cdot \hat{y}_w$$

$\hat{y}_w$

۲) $\omega = o$: $\frac{\partial J}{\partial u_w} = -v_c + v_c \cdot \hat{y}_w$

سوال چهارم) مشتقات جزئی Jnaive-softmax(vc,o,U) نسبت به U.

$$\frac{\partial J(v_c, o, U)}{\partial U} = v_c(\hat{y} - y)^T = \left[\frac{\partial J}{\partial u_1} \; \frac{\partial J}{\partial u_2} \; \frac{\partial J}{\partial u_3} \cdots \frac{\partial J}{\partial u_{|Vocab|}}\right]$$

سوال پنجم) مشتق Leaky ReLU نسبت به x:

$0 < \alpha < 1$

$$f(x) = \max(\alpha x, x) \begin{cases} \alpha x \; ; \; x < 0 & (۱ \\ x \; ; \; x \geq 0 & (۲ \end{cases}$$

۱) $x < 0$: $\frac{d f(x)}{dx} = \frac{d(\alpha x)}{dx} = \alpha$

در $x = 0$ مشتق نداریم چون مشتق‌پذیر نیست

۲) $x \geq 0$: $\frac{d f(x)}{dx} = \frac{d(x)}{dx} = 1$

و به طور دلخواه ۱ را فرض کردیم.

$$\Longrightarrow f'(x) = \begin{cases} \alpha \; ; \; x < 0 \\ 1 \; ; \; x \geq 0 \end{cases}$$

سوال ششم) مشتق sigmoid نسبت به x:

$$\sigma = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

$$\sigma'(x) = \frac{e^x(e^x+1) - e^x e^x}{(e^x+1)^2} = \overbrace{\left(\frac{e^x}{e^x+1}\right)}^{\sigma(x)}\left(\overbrace{\frac{e^x+1}{e^x+1}}^{1} - \overbrace{\frac{e^x}{e^x+1}}^{\sigma(x)}\right) = \sigma(x)(1-\sigma(x))$$

سوال هفتم)(i) تکرار سوال دو و سه برای Jneg-sample:

$$\frac{\partial J_{neg\text{-}sample}}{\partial v_c} = -\frac{\partial}{\partial v_c}\left(\log(\sigma(u_o^T \cdot v_c)) - \sum_{k=1}^{K} \log(\sigma(-u_k^T \cdot v_c))\right)$$

$$= (\sigma(u_o^T \cdot v_c) - 1)\, u_o + \sum_{k=1}^{K} (1-\sigma(-u_k^T \cdot v_c))\, u_k = (\sigma(u_o^T \cdot v_c)-1)\, u_o + \sum_{k=1}^{K} \sigma(u_k^T v_c)\, u_k$$

$$\frac{\partial J_{neg\text{-}sample}}{\partial u_o} = \frac{\partial}{\partial u_o}\left(\log(\sigma(u_o^T \cdot v_c)) - \sum_{k=1}^{K} \log(\sigma(-u_k^T \cdot v_c))\right)$$

$$= -\frac{\cancel{\sigma(u_o^T \cdot v_c)} \cdot (1-\sigma(u_o^T \cdot v_c))}{\cancel{\sigma(u_o^T \cdot v_c)}} \cdot v_c = -(1-\sigma(u_o^T \cdot v_c))\, v_c = (-1+\sigma(u_o^T v_c))\, v_c$$

$$\frac{\partial J_{neg\text{-}sample}}{\partial u_k} = -\frac{\partial}{\partial u_k}\left(\log \sigma(u_o^T \cdot v_c) - \sum_{k=1}^{K} \log \sigma(-u_k^T \cdot v_c)\right)$$

$$= 0 - \frac{\partial}{\partial u_k} \sum_{k=1}^{K} \log(\sigma(-u_k^T \cdot v_c)) = -\frac{\cancel{\sigma(-u_k^T \cdot v_c)} \cdot (1-\sigma(-u_k^T \cdot v_c)) \cdot (-v_c)}{\cancel{\sigma(-u_k^T \cdot v_c)}}$$

$$= +(1-\sigma(-u_k^T \cdot v_c))\, v_c$$

سوال هفتم)(ii): $(\sigma(u_o^T v_c))$ عبارت مشترک در هر سه مشتق جزئی است. پس میتوانیم آن را یکبار در مشتق جزئی J نسبت به v_c حساب کنیم و ذخیره کنیم و در محاسبه ی دو مشتق جزئی دیگر از آن استفاده کنیم.

سوال هفتم)(iii) :همانطور که می بینیم، بیشترین تکراری که باید انجام دهیم [1.. K]است، اما در سافت مکس ساده باید روی تمام واژگان تکرار کنیم. در واقع محاسبه ی مخرج سافت مکس در J ساده هزینه ی محاسباتی زیادی برای *vocabulary* بزرگ دارد. بخاطر همین کلماتی میدانیم که در *context* کلمه ی مرکزی هستند را جدا گانه حساب میکنیم و به صورت رندوم از بین کلماتی که میدانیم در *context* کلمه ی مرکزی نیستند انتخاب کرده و با فرمول جداگانه حساب میکنیم تا هزینه ی محاسباتی و زمانی و حافظه ای کمتری برایمان داشته باشد.

سوال هشتم)

$$s \in S \; ; \; w_s = w_k \Rightarrow u_k = u_s$$

$$\frac{\partial}{\partial u_k} J_{neg-sample} = -\frac{\partial}{\partial} \left(\log \sigma(u_o^T \cdot v_c) - \sum_{k=1}^{K} \log \sigma(-u_k^T \cdot v_c) \right)$$

$$= -\sum_{s=1}^{|S|} \left(\frac{1}{\sigma(-u_k^T \cdot v_c)} \right) \left(\frac{\partial}{\partial u_k} \sigma(-u_k^T v_c) \right) = -\sum_{s=1}^{|S|} \frac{\sigma(-u_k^T v_c)(1-\sigma(u_k^T \cdot v_c)) \cdot (-v_c)}{\sigma(-u_k^T \cdot v_c)}$$

$$= \sum_{s=1}^{|S|} (1-\sigma(u_k^T \cdot v_c)) \cdot v_c = |S| \left(1-\sigma(u_k^T \cdot v_c)\right) v_c$$

سوال هشتم (ii))

$$\frac{\partial J_{skip-gram}}{\partial U} (v_c, w_{t-m}, \ldots, w_{t+m}, U) = \frac{\partial}{\partial U} \sum J(v_c, w_{t+j}, U)$$

$$= \sum_{\substack{-m \le j \le m \\ j \ne 0}} \frac{\partial}{\partial U} J(v_c, w_{t+j}, U)$$

$$\frac{\partial J_{skip-gram}}{\partial v_c} (v_c, w_{t-m}, \ldots, w_{t+m}, U) = \frac{\partial}{\partial v_c} \sum_{\substack{-m \le j \le m \\ j \ne 0}} J(v_c, w_{t+j}, U)$$

$$= \sum_{\substack{-m \le j \le m \\ j \ne 0}} \frac{\partial}{\partial v_c} J(v_c, w_{t+j}, U)$$

$$\frac{\partial J_{skip-gram}}{\partial v_w} (v_c, w_{t-m}, \ldots, w_{t+m}, U) = \frac{\partial}{\partial v_w} \sum_{\substack{-m \le j \le m \\ j \ne 0}} J(v_c, w_{t+j}, U)$$

$$= \frac{\partial}{\partial v_w} J(v_c, w, U)$$