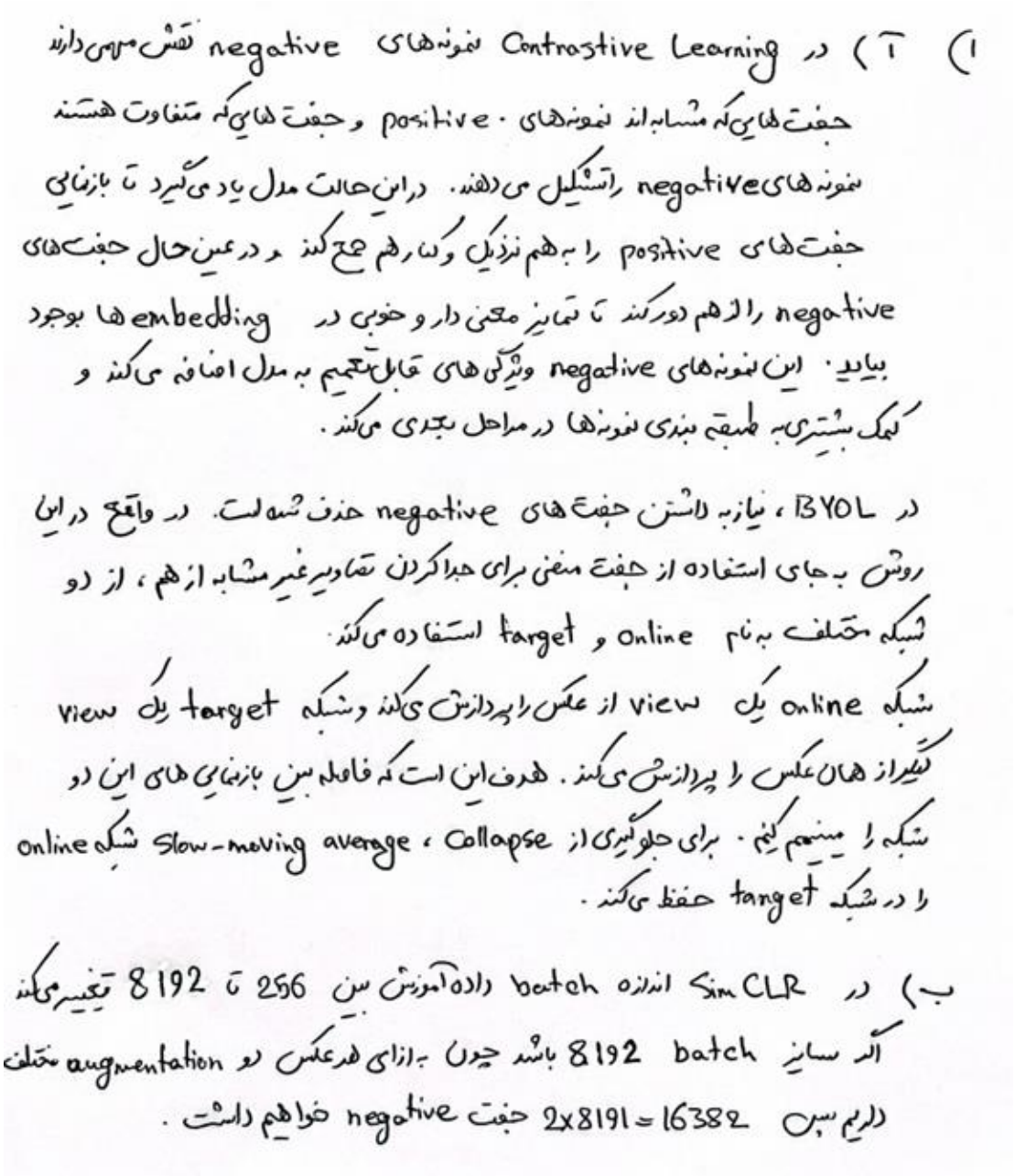
سوال یک

1)



ادامه ب) اما آموزش مدل با سایز batch بزرگ ممکن است باعث unstable شدن
مدل هنگام استفاده از SGD/Momentum با scaling خطی نرخ یادگیری
می‌شود. برای پایدار کردن آموزش از بهینه‌ساز LARS به ازای هم
سایز batch های مختلف استفاده می‌شود.

2)

آ) شبکه teacher از همان ابتدا از پارامترهای شبکه student به عنوان مرجع استفاده می‌کند و به مرور زمان با میانگین‌گیری نمایی (EMA) به‌روزرسانی می‌شود. این میانگین‌گیری نمایی باعث می‌شود که شبکه teacher نسبت به تغییرات سریع و نوسانات شبکه student مقاومت بیشتری داشته باشد و بتواند بازنمایی‌های پایدارتر و بهتر ارائه دهد. این پایداری و کیفیت بهتر در بازنمایی‌های شبکه teacher کمک می‌کند تا دانش بهتری به شبکه student منتقل شود و در نتیجه، شبکه student بتواند بازنمایی‌های معنادارتر و دقیق‌تری یاد بگیرد.

ب) در فرآیند آموزشDINO ، از تکنیک‌های مختلفی برای تشویق شبکه teacher به تمرکز بیشتر بر روی اشیاء اصلی تصویر و نادیده گرفتن پس‌زمینه استفاده می‌شود. یکی از این تکنیک‌ها استفاده از تغییرات گسترده و شدید در تصاویر ورودی (augmentation)است که شامل برش، چرخش، تغییر رنگ و کنتراست می‌شود. این تغییرات باعث می‌شوند شبکه مجبور شود روی ویژگی‌های پایدار و اصلی تمرکز کند. همچنین، روش یادگیری خودنظارتی(self-supervised learning) به شبکه کمک می‌کند تا بازنمایی‌هایی که به طور خودکار برای تشخیص و تمایز اشیاء مفید هستند را یاد بگیرد.

ج) رای جلوگیری از collapse، یعنی زمانی که شبکه به بازنمایی‌های بی‌معنی و تکراری می‌رسد، راهکارهای مختلفی ارائه شده است:

- DINO: از یک مکانیسم نرمال‌سازی استفاده می‌شود تا مطمئن شوند که بردارهای بازنمایی طول یکسانی دارند. همچنین، مکانیزم centering در شبکه teacher به کار گرفته می‌شود تا از خروجی‌های بسیار مشابه جلوگیری کند.
- DINOv2: در این نسخه، تکنیک‌های بیشتری برای جلوگیری از collapse به کار گرفته شده که شامل استفاده از انواع lossهای پیچیده‌تر و همچنین استفاده از augmentations پیشرفته‌تر برای افزایش تنوع بازنمایی‌ها است.

د) در پیاده‌سازیDINOv2 ، نکات زیر رعایت شده است:

- **استفاده از augmentations پیشرفته**: برای افزایش تنوع داده‌های ورودی و بهبود بازنمایی‌ها، از augmentations پیچیده‌تری استفاده شده است.
- **تنظیم دقیق هایپرپارامترها**: هایپرپارامترها با دقت بالاتری تنظیم شده‌اند تا عملکرد بهینه شبکه تضمین شود.
- **بهبود ساختار شبکه**: تغییرات و بهبودهایی در ساختار شبکه و توابع هزینه (loss functions) اعمال شده تا کارایی افزایش یابد و از collapse جلوگیری شود.
- **استفاده از روش‌های بهینه‌سازی بهتر**: از روش‌های بهینه‌سازی پیشرفته‌تری برای بهبود عملکرد شبکه استفاده شده است.

ه) تفاوت‌های اصلی بین DINO شامل موارد زیر می‌شود:

- **مکانیزم EMA**: درDINO، شبکه teacher به صورت EMA از شبکه student به‌روزرسانی می‌شود، در حالی که در BYOL این مکانیزم به این شکل پیاده‌سازی نشده است.

- **استفاده از centering**: DINO از یک مکانیزم centering در شبکه teacher برای جلوگیری از collapse استفاده می‌کند، در حالی که BYOL از این مکانیزم استفاده نمی‌کند.
- **استفاده از augmentations متفاوت**DINO: از augmentations پیشرفته‌تری استفاده می‌کند که باعث بهبود بازنمایی‌ها می‌شود.

این تفاوت‌ها باعث شده‌اند که DINO نتایج بهتری نسبت به BYOL به دست آورد. مکانیزم EMA باعث پایداری بیشتر بازنمایی‌ها و در نتیجه آموزش بهتر شبکه student می‌شود. استفاده از centering نیز به جلوگیری از collapse کمک کرده و بازنمایی‌های متنوع‌تری ایجاد می‌کند. augmentations پیشرفته‌تر نیز باعث شده‌اند که شبکه روی ویژگی‌های پایدارتر و مهم‌تر تمرکز کند.

سوال دو

1)

آ) معتبر است.

این تابع نسبت به ترتیب همسایگان invariant است. چون میانگین همسایگان را حساب می‌کند پس جابجایی گره‌ها تاثیری در خروجی نخواهد داشت. از آنجایی که هر دو بخش تابع یا مستقل از ترتیب همسایه یا نسبت به جایگشت invariant هستند، کل تابع هم equivariant است.

ب) معتبر نیست.

تابع max بطور کلی نسبت به ترتیب همسایه‌ها invariant است. اما در این تابع چون وزن‌های مختلفی به هر همسایه داده شده پس با تغییر ترتیبشان خروجی تغییر خواهد کرد و این تابع invariant نیست. پس بدلیل تفاوت خروجی و نداشتن نظم خروجی با تغییر ورودی، این تابع equivariant نخواهد بود.

ج) معتبر است.

همانطور که در بخش قبلی گفته شد، تابع max نسبت به ترتیب همسایه‌ها invariant است. همچنین چون وزن همگی همسایه‌ها یکسان است پس این تابع نیز invariant خواهد بود و با تعویض جایگاه همسایه‌ها خروجی تغییر نخواهد کرد. چون این تابع هم equivariant است.

$$G = (V, E)$$

$$h_v^{(t+1)} = \sigma\left(W h_v^{(t)} + \sum_{u \in N(v)} W' h_u^{(t)} \right)$$

اگر این جایگشت π با ماتریس جایگشت P در نظر بگیریم :

میدانیم که در ابتدای کار رابطه روبرو برقرار است :

$$H_P^{(0)} = P H^{(0)}$$

یعنی وزن‌های اولیه $H^{(0)}$ با ضرب در ماتریس جایگشت به $H_P^{(0)}$ تبدیل می‌شود و برای این کار نیاز به جابجایی گره‌ها (وزن‌ها) با ماتریس P داریم.

همچنین اگر $H_P^{(t)} = P H^{(t)}$ باشد و همچنین طبق جایگشت، ماتریس همسایه‌ها

$$\circledast \quad A_P = P A P^T$$

بصورت روبه رو محاسبه شود، داریم:

$$\underbrace{h_{\pi(v)}^{(t+1)}}_{H_P^{(t+1)}} = \sigma\left(W \overbrace{h_{\pi(v)}^{(t)}}^{H_P^{(t)}} + \overbrace{\sum_{u \in N(\pi(v))} W' h_u^{(t)}}^{A_P} \right)$$

$$H_P^{(t+1)} = \sigma\left(W(P H^{(t)}) + A_P W'(P H^{(t)}) \right)$$

$$H_P^{(t+1)} = \sigma\left(W P H^{(t)} + P A P^T W' P H^{(t)} \right) \xrightarrow{P P^T = I}$$

$$H_P^{(t+1)} = \sigma\left(P(W H^{(t)} + A W' H^{(t)}) \right)$$

$$H_P^{(t+1)} = P\, \sigma\left(W H^{(t)} + A W' H^{(t)} \right) = P H^{(t+1)}$$

پس اگر $H_P^{(t)} = P H^{(t)}$ باشد آنگاه $H_P^{(t+1)} = P H^{(t+1)}$ نیز برقرار خواهد بود. چون در ابتدا $H_P^{(0)} = P H^{(0)}$ است پس بعد از t تکرار نیز $H_P^{(t+1)} = P H^{(t+1)}$ برقرار است.

سوال سه

1) آشفتگی‌های خصمانه فراگیر، آشتفگی‌ای در تشخیص تصویر هستند که وقتی به تصاویر طبیعی اضافه می‌شوند، باعث می‌شوند طبقه‌بندی‌کننده‌های پیشرفته شبکه عصبی عمیق آن تصاویر را با احتمال بالا طبقه‌بندی اشتباه کنند.

2) کشف آشفتگی‌های خصمانه فراگیر به چند دلیل زیر مهم است:

- امنیت: این آشفتگی ها آسیب پذیری های قابل توجهی را در شبکه های عصبی عمیق نشان می دهد که می تواند توسط دشمنان در برنامه های کاربردی دنیای واقعی مورد سوء استفاده قرار گیرد.
- generalization: اینکه یک آشفتگی می تواند چندین تصویر و حتی مدل های مختلف را فریب دهد، نشان می دهد که الگوها و ساختارهای مشترکی در مرزهای تصمیم گیری شبکه های عصبی وجود دارد. این بینش می‌تواند به درک عمیق‌تری از نحوه تصمیم‌گیری شبکه‌های عصبی و چگونگی بهبود آنها منجر شود.
- بهبود مدل های هوش مصنوعی: با کشف این آشفتگی‌ها، می‌توان دفاع بهتری در برابر حملات متخاصم ایجاد کرد. این شامل ایجاد روش‌های آموزشی‌ای است که مدل‌ها را در برابر چنین آشفتگی‌هایی مقاوم‌تر می‌کند.
- درک بهتر مرزهای تصمیم: وجود این آشفتگی‌ها همبستگی‌های هندسی مهمی را در بین مرزهای تصمیم گیری با ابعاد بالا طبقه بندی‌کننده‌ها نشان می دهد. این موضوع می‌تواند به اصلاح مدل‌های ریاضی شبکه‌های عصبی و بهبود دقت و قابلیت اطمینان آنها کمک کند.

3)

اگر بردار آشفتگی را برابر v در نظر بگیریم آنگاه دیتاست perturbed شده بصورت زیر خواهد شد:

$$D' = \{x + v | x \in D\}$$

در این حالت باید مقدار تابع g را روی دیتاست D' ماکزیمم کنیم بطوریکه نرم p بردار v کوچکتر از مقدار اپسیلون شود:

$$\max_{v} g(D')$$

$$subject\ to \quad \|v\|_p < \epsilon$$

ϵ حداکثر نُرم آشفتگی مجاز است.

سوال چهار

1)

شباهت‌ها:

چندوجهی بودن(Multimodal Nature):

هر سه مدل برای پردازش و تحلیل داده‌های چندوجهی طراحی شده‌اند، یعنی داده‌هایی که هم شامل متن و هم شامل تصویر هستند. این مدل‌ها قابلیت یادگیری همزمان از داده‌های تصویری و متنی را دارند که باعث می‌شود بتوانند وظایفی مانند تطبیق تصویر و متن، توصیف تصاویر و جستجوی متنی-تصویری را به خوبی انجام دهند.

استفاده از معماری‌های ترانسفورمر (Transformer Architectures):

هر سه مدل از معماری ترانسفورمر برای پردازش داده‌ها استفاده می‌کنند. ترانسفورمرها به دلیل قابلیت‌هایشان در یادگیری توالی‌ها و مدیریت روابط طولانی‌مدت بین داده‌ها بسیار موثر هستند و این مدل‌ها نیز با استفاده از این معماری توانسته‌اند عملکرد قابل توجهی در وظایف چندوجهی داشته باشند.

تفاوت‌ها:

روش پیش‌آموزش (Pre-training Method):

- SimVLM: این مدل از یک روش پیش‌آموزش ساده و مقیاس‌پذیر به نام "Simple Visual Language Model" استفاده می‌کند که به وسیله آموزش بر روی داده‌های متنی و تصویری بزرگ، قابلیت‌های چندوجهی را کسب می‌کند.
- CLIP: در CLIP، از روش آموزش متضاد استفاده می‌شود که در آن جفت‌های متن-تصویر مرتبط و غیرمرتبط به مدل داده می‌شود و مدل باید بتواند جفت‌های مرتبط را تشخیص دهد.
- CoCa: CoCa از یک روش پیش‌آموزش چندوجهی استفاده می‌کند که به صورت مشترک بر روی داده‌های تصویری و متنی آموزش می‌بینند، اما به طور خاص بر روی تولید و تطبیق متن و تصویر به طور همزمان تمرکز دارد.

معماری و نحوه تعامل بین متن و تصویر:

- SimVLM: از یک مدل ترانسفورمر دوتایی (dual-transformer) استفاده می‌کند که یکی برای پردازش متن و دیگری برای پردازش تصویر به کار می‌رود و سپس این دو با هم ترکیب می‌شوند.
- CLIP: در CLIP دو ترانسفورمر مجزا برای پردازش متن و تصویر به کار می‌رود و سپس با استفاده از یک فضای تعبیه مشترک(joint embedding space) ، این دو نوع داده به هم مرتبط می‌شوند.
- CoCa: CoCa از یک معماری یکپارچه ترانسفورمر استفاده می‌کند که همزمان داده‌های متنی و تصویری را پردازش می‌کند و از این رو توانایی تعامل پیچیده‌تری بین متن و تصویر دارد.

2)

آ) ماژول‌های اصلی معماری CLIP که دارای مدل هستند، image encoder و text encoder هستند.

1. Image encoder:

این ماژول تصاویر را در یک بردار با اندازه ثابت (embedding) پردازش و encode می‌کند که ویژگی‌های برجسته تصاویر را ثبت می‌کند.

انکودر تصویر در CLIP معمولاً Vision Transformer (ViT) یا ResNet است. ViT با اعمال یک ترانسفورمر مستقیماً روی دنباله‌های پچ‌های تصویری عمل می‌کند، در حالی که ResNet نوعی شبکه عصبی کانولوشنال (CNN) است که از یادگیری residual برای سهولت و عملکرد بهتر آموزش شبکه‌های عمیق استفاده می‌کند.

2. Text encoder:

این ماژول توضیحات متن را در یک بردار با اندازه ثابت (embedding) پردازش و encoder می‌کند که تفسیر معنایی متن را نمایش می دهد.

انکودر متن در CLIP یک مدل مبتنی بر ترانسفورمر است که مشابه مدل هایی مانند GPT (ترانسفورماتور از پیش آموزش داده شده تولیدی) استفاده می شود. ورودی متن را با استفاده از مکانیسم‌های توجه پردازش می‌کند تا نمایشی متراکم از متن ایجاد کند.

در انتها توسط الگوریتم contrastive learning که لاسی از گونه InfoNSE loss دارد استفاده میشود تا فضای امبدینگ متن و تصویر را به همدیگر نزدیک کند.

ب)

$$\begin{cases} L_1 = -\frac{1}{N} \log \frac{e^{sim(x_i, y_i)/\tau}}{\sum_{j=1}^{N} e^{sim(x_i, y_j)/\tau}} \\ S_{i,j} = \frac{x_i \cdot y_j}{\tau \|x_i\| \, \|y_j\|} \quad , \quad S_{i,j} = \frac{sim(x_i, y_j)}{\tau} \end{cases}$$

$$\Rightarrow L_1 = -\frac{1}{N} \log \frac{e^{S_{i,j}}}{\sum_{j=1}^{N} e^{S_{i,j}}}$$

اثبات در صفحه بعد.

$$\frac{\partial L_1}{\partial x_i} = \sum_{j=1}^{N} \frac{\partial L_1}{\partial s_{i,j}} \times \frac{\partial s_{i,j}}{\partial x_i} \quad = ?$$

recall that:

$$\frac{\partial \log u(x)}{\partial x} = \frac{u'(x)}{u}$$

$$\frac{\partial L_1}{\partial s_{i,j}} = -\frac{1}{N} \frac{\frac{\partial}{\partial s_{i,j}}\left(\frac{e^{s_{i,j}}}{\sum e^{s_{i,j}}}\right)}{\frac{e^{s_{i,j}}}{\sum_{j=1}^{N} e^{s_{i,j}}}}$$

$$\Rightarrow \frac{\partial L_1}{\partial s_{i,j}} = -\frac{1}{N} \frac{e^{s_{i,j}} \times \sum_{j=1}^{N} e^{s_{i,j}} - e^{s_{i,j}} \times e^{s_{i,j}}}{\left(\sum_{j=1}^{N} e^{s_{i,j}}\right)^2} \frac{e^{s_{i,j}}}{\sum_{j=1}^{N} e^{s_{i,j}}} = \frac{-1}{N}\left(1 - \frac{e^{s_{i,j}}}{\sum_{j=1}^{N} e^{s_{i,j}}}\right)$$

$$\frac{\partial s_{i,j}}{\partial x_i} = \frac{y_j \cdot \tau \|x_i\| \cdot \|y_j\| - x_i \cdot y_j \cdot \frac{x_i}{\|x_i\|}}{\left(\tau \|x_i\| \|y_i\|\right)^2}$$

$$\frac{\partial L_1}{\partial x_i} = \frac{-1}{N} \sum_{j=1}^{N} \left(1 - \frac{e^{s_{i,j}}}{\sum_{j=1}^{N} e^{s_{i,j}}}\right) \left(\frac{y_j \tau \|x_i\| \cdot \|y_i\| - x_i \cdot y_j \frac{x_i}{\|x_i\|}}{\left(\tau \|x_i\| \|y_i\|\right)^2}\right)$$