



## تحلیل هوشمند تصاویر زیست پزشکی

نیم سال اول ۰۴-۰۳

مدرس: محمدحسین رهبان

### تمرین چهارم

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- در طول ترم امکان ارسال با تاخیر تمرین ها بدون کسر نمره تا سقف ۱۲ روز وجود دارد. محل بارگزاری جواب تمرین ها بعد از ۴ روز بسته خواهد شد و پس از گذشت این مدت، پاسخ های ارسال شده پذیرفته نخواهند شد.
- توجه داشته باشید که نوت بوک های شما باید قابلیت باز اجرای ۱۰۰ درصد داشته باشند و در صورت نیاز به نصب یک کتابخانه یا دسترسی به یک فایل، مراحل نصب و دانلود (از یک محل عمومی) در نوت بوک وجود داشته باشد.
- هم فکری در انجام تمرین مانعی ندارد، فقط توجه داشته باشید که پاسخ تمرین حتما باید توسط خود شخص نوشته شده باشد. همچنین در صورت هم فکری در هر تمرین، در ابتدای جواب تمرین نام افرادی که با آن ها هم فکری کرده اید را حتما ذکر کنید.
- برای پاسخ به سوالات نظری در صورتی که از برگه خود عکس تهیه می کنید، حتما توجه داشته باشید که تصویر کاملاً واضح و خوانا باشد. در صورتی که خوانایی کافی را نداشته باشد، تصحیح نخواهد شد.
- محل بارگذاری سوالات نظری و عملی در هر تمرین مجزا خواهد بود. به منظور بارگذاری بایستی تمارین تئوری در یک فایل زیپ با نام IABI\_Theo\_hw4\_[First-Name]\_[Last-Name]\_[Student-Id].zip و تمارین عملی نیز در یک فایل مجزای زیپ با نام IABI\_Prac\_hw4\_[First-Name]\_[Last-Name]\_[Student-Id].zip بارگذاری شوند.
- در صورت وجود هرگونه ابهام یا مشکل، در کوثرای درس آن مشکل را بیان کنید و از پیغام دادن مستقیم به دستیاران آموزشی خودداری کنید.

### بخش نظری (۵۰ نمره)

## سوال اول: تابع Aggregation در MIL (۱۰ نمره)

(الف) توضیح دهید که چرا انتخاب توابع aggregation با خاصیت permutation invariance برای مسائل MIL ضروری است.

(ب) در مسئله یادگیری (MIL) Multiple Instance Learning، سه تابع aggregation برای ترکیب ویژگی های پیچ ها در نظر گرفته شده اند:

$$f_{\max}(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n) \quad (\text{آ})$$

$$f_{\text{prod}}(x_1, x_2, \dots, x_n) = \prod_{i=1}^n x_i \quad (\text{ب})$$

$$f_{\text{concat}}(x_1, x_2, \dots, x_n) = [x_1, x_2, \dots, x_n] \quad (\text{ج})$$

برای هر یک از این توابع توضیح دهید که آیا مناسب برای مسائل MIL هستند یا نه و چرا.

(الف) در مسائل Multiple Instance Learning (MIL) یک نمونه معمولاً شامل مجموعه‌ای از پچ‌ها یا نمونه‌های فرعی است. هدف این است که مدل از ویژگی‌های این پچ‌ها برای پیش‌بینی‌های کلی استفاده کند. در اینجا ترتیب پچ‌ها اهمیت ندارد، زیرا هیچ‌گونه اطلاعات اضافی در مورد ترتیب پچ‌ها وجود ندارد. بنابراین، تغییر در ترتیب پچ‌ها نباید بر خروجی مدل تأثیر بگذارد.

اگر تابع aggregation انتخابی خاصیت permutation invariance نداشته باشد، تغییر در ترتیب پچ‌ها ممکن است باعث تغییرات غیر منطقی در پیش‌بینی‌های مدل شود. این امر می‌تواند مدل را از دقت صحیح باز دارد و پیش‌بینی‌هایی که تنها به ترتیب پچ‌ها وابسته هستند، به دست دهد. به همین دلیل، انتخاب توابع aggregation با خاصیت permutation invariance برای مسائل MIL ضروری است.

(ب) در مسئله MIL، سه تابع aggregation به صورت زیر تعریف شده‌اند:

$$f_{\max}(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n) \quad (\text{آ})$$

$$f_{\text{prod}}(x_1, x_2, \dots, x_n) = \prod_{i=1}^n x_i \quad (\text{ب})$$

$$f_{\text{concat}}(x_1, x_2, \dots, x_n) = [x_1, x_2, \dots, x_n] \quad (\text{ج})$$

**تابع  $f_{\max}$ :** این تابع تنها بزرگترین مقدار از مجموعه پچ‌ها را انتخاب می‌کند. در مسائل MIL، این نوع aggregation می‌تواند مفید باشد زیرا اگر یکی از پچ‌ها ویژگی مهم و برجسته‌ای داشته باشد، این ویژگی به طور مؤثر در پیش‌بینی نهایی مدل تأثیر می‌گذارد. به طور مثال، اگر یکی از پچ‌ها در یک تصویر حاوی اطلاعات حیاتی باشد (مثلاً وجود یک شیء خاص)، این تابع می‌تواند به درستی آن اطلاعات را در پیش‌بینی نهایی وارد کند. همچنین این تابع خاصیت permutation invariance دارد.

**نتیجه:** تابع  $f_{\max}$  معمولاً برای MIL مناسب است.

**تابع  $f_{\text{prod}}$ :** این تابع ضرب ویژگی‌های پچ‌ها را محاسبه می‌کند. در حالی که این تابع از نظر ریاضی خاصیت permutation invariance دارد، معمولاً برای مسائل MIL مناسب نیست. دلیل این امر این است که اگر یکی از پچ‌ها ویژگی کوچکی (مثلاً نزدیک به صفر) داشته باشد، ضرب تمام ویژگی‌ها ممکن است منجر به صفر یا مقدار بسیار کوچکی شود که پیش‌بینی مدل را به شدت تحت تأثیر قرار می‌دهد. همچنین، این نوع aggregation ممکن است برای داده‌هایی که ویژگی‌هایشان مقادیر پراکنده و متفاوت دارند، مناسب نباشد.

**نتیجه:** تابع  $f_{\text{prod}}$  معمولاً برای MIL مناسب نیست، مگر در موارد خاص که ویژگی‌ها مقادیر بزرگ یا مساوی یک داشته باشند و از نظر مدل این نوع aggregation منطقی باشد.

**تابع  $f_{\text{concat}}$ :** این تابع ویژگی‌های پچ‌ها را به صورت یک بردار متصل می‌کند. از آنجا که تابع concat ترتیب پچ‌ها را در نظر می‌گیرد، خاصیت permutation invariance ندارد و با تغییر ترتیب پچ‌ها، خروجی تغییر می‌کند. به همین دلیل، استفاده از این تابع در مسائل MIL معمولاً توصیه نمی‌شود، زیرا تغییرات در ترتیب پچ‌ها می‌تواند پیش‌بینی‌های مدل را تحت تأثیر قرار دهد. این تابع بیشتر برای استفاده در مدل‌هایی که ترتیب ورودی‌ها اهمیت دارد (مثل شبکه‌های عصبی بازگشتی) مناسب است.

**نتیجه:** تابع  $f_{\text{concat}}$  برای MIL مناسب نیست، زیرا خاصیت permutation invariance ندارد و ترتیب پچ‌ها را در نظر می‌گیرد.

## سوال دوم: طبقه‌بندی رابطه‌ای در گراف (۶ نمره)

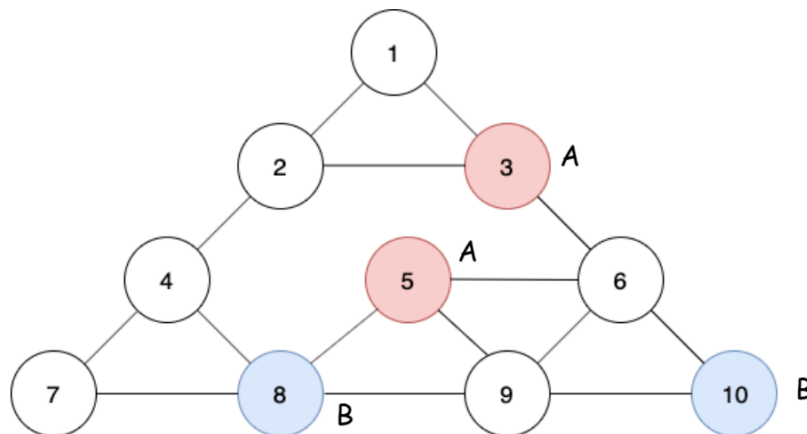
می‌خواهیم از طبقه‌بندی رابطه‌ای برای پیش‌بینی برجسب‌های گره‌ها استفاده کنیم. گراف  $G$  را در نظر بگیرید که در شکل ۱ نشان داده شده است. می‌خواهیم گره‌ها را به دو کلاس  $A$  و  $B$  طبقه‌بندی کنیم. برجسب‌های گره‌های ۳، ۵، ۸ و ۱۰ داده شده‌اند.

استفاده از یک طبقه‌بند رابطه‌ای احتمالاتی برای پیش‌بینی  $Y_i$  برای گره  $i$  به صورت زیر تعریف می‌شود:

$$P(Y_i = c) = \frac{1}{|N_i|} \sum_{(i,j) \in E} W(i,j) P(Y_j = c)$$

که در آن  $|N_i|$  تعداد همسایه‌های گره  $i$  است.

- فرض کنید همه‌ی یال‌ها وزن ۱ دارند.
- برای گره‌های دارای برجسب، با مقدار واقعی  $Y$  مقاداردهی اولیه کنید و برای گره‌های بدون برجسب، از مقداردهی اولیه بی‌طرفانه یعنی  $P(Y_i = A) = 0.5$  استفاده کنید.
- گره‌ها را به ترتیب شماره گره به‌روزرسانی کنید.



شکل ۱: گراف  $G$

(آ) پس از مرحله (تکرار) دوم، مقدار  $P(Y_i = B)$  را برای  $i = 2, 4, 6$  بدست آورید.

(ب) اگر از ۰.۵ به عنوان آستانه احتمال استفاده کنیم، یعنی گره  $i$  به کلاس  $A$  تعلق دارد اگر  $P(Y_i = A) > 0.5$ ، کدام گره در پایان به کلاس  $A$  تعلق خواهد داشت؟ و کدام به کلاس  $B$  تعلق خواهد داشت؟

پاسخ:

(آ) مقادیر به‌دست‌آمده: (به ترتیب ۲، ۴، ۶) ۰.۲۴، ۰.۶۷، ۰.۴۶.

(ب) در نهایت، مقادیر  $P(Y_i = A)$  به صورت زیر خواهند بود:

(۱ : ۰.۸۶), (۲ : ۰.۷۱), (۳ : ۱.۰۰), (۴ : ۰.۲۹), (۵ : ۱.۰۰), (۶ : ۰.۵۲), (۷ : ۰.۱۴), (۸ : ۰.۰۰), (۹ : ۰.۳۸), (۱۰ : ۰.۰۰)

بنابراین گره‌های با شماره‌های [۱، ۲، ۳، ۵، ۶] به کلاس  $A$  تعلق دارند و گره‌های با شماره‌های [۴، ۷، ۸، ۹، ۱۰] به کلاس  $B$  تعلق دارند.

## سوال سوم: تابع Aggregator در GNN (۱۲ نمره)

فرض کنید یک گراف  $G$  داریم. ماتریس  $A$  ماتریس مجاورت آن است که نشان می‌دهد کدام گره‌ها به هم متصل‌اند و همچنین ماتریس قطری  $D$  که به ابعاد  $|V| \times |V|$  است، ماتریس درجه گره‌های گراف است. بطوریکه عناصر قطر آن، درجه هر گره را نشان می‌دهد. فرض کنید  $h_i^{(l)}$  نمایش‌دهنده Embedding گره  $i$  در لایه  $l$  باشد. همچنین از یک تجمیع‌کننده میانگین‌گیر (Mean Aggregator) برای آپدیت گره‌ها استفاده می‌کنیم به طوری که:

$$h_i^{(l+1)} = \frac{1}{|N_i|} \sum_{j \in N_i} h_j^{(l)}$$

اگر مقدار Embedding همه گره‌ها در لایه  $L$  را به شکل زیر Vectorize کنیم:

$$H^{(L)} = \begin{bmatrix} h_1^{(L)} \\ h_2^{(L)} \\ \vdots \\ h_{|V|}^{(L)} \end{bmatrix}$$

(آ) یک رابطه ماتریسی پیدا کنید که با داشتن ماتریس‌های  $A$ ،  $D$  و بردار  $H^{(L)}$ ، بردار  $H^{(L+1)}$  که هر گره با استفاده از تابع تجمیع‌کننده میانگین‌گیر آپدیت شده را محاسبه کند.

(ب) حال فرض کنید در گراف  $G$  می‌خواهیم بصورت تصادفی با احتمال یکنواخت (Uniform) حرکت کنیم. یعنی اگر گره  $i$  به گره‌های  $j$  و  $k$  متصل باشد، به احتمال مساوی از گره  $i$  به گره  $j$  یا  $k$  می‌توانیم برویم. ماتریس انتقال (Transition) را برای این حرکت تصادفی با داشتن ماتریس‌های  $A$  و  $D$  بدست بیاورید. با توجه به پاسخ بخش «آ» و این بخش چه نتیجه‌ای می‌توان گرفت؟  
(راهنمایی: ماتریس Transition  $|V| \times |V|$  است که درایه  $(i, j)$  در آن احتمال رفتن از گره  $i$  به گره  $j$  را نشان می‌دهد.)

(ج) حال فرض کنید یک Skip Connection به تابع Aggregator اضافه کنیم که مقدار Embedding قبلی خودش نیز در مقدار Embedding لایه بعدش تاثیرگذار باشد، به طوریکه:

$$h_i^{(l+1)} = \frac{1}{3} h_i^{(l)} + \frac{2}{3} \frac{1}{|N_i|} \sum_{j \in N_i} h_j^{(l)}$$

همانند بخش «آ» رابطه ماتریسی مربوط به این تابع تجمیع‌کننده را بنویسید.

(د) همانطور که تا اینجا با تابع تجمیع‌کننده Mean آشنا شده‌اید، توابع تجمیع‌کننده دیگری برای تجمیع Embedding همسایه‌های یک گره در گراف وجود دارند. دو تابع دیگر عبارت است از:

$$h_v^{l+1} = \text{AGGREGATE}_{\min} (\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}) = \min_{u \in \mathcal{N}(v)} h_u^{(l)}$$

و

$$h_v^{l+1} = \text{AGGREGATE}_{\text{sum}} (\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}) = \sum_{u \in \mathcal{N}(v)} h_u^{(l)}$$

حال یک گراف  $G(V, E)$  بیابید که دارای دو شرط زیر باشد:

- $|V| \geq 3$
- $\forall v \in V, \deg(v) \geq 2$

و همچنین مقدار اولیه‌ای به عنوان Embedding (همان  $h_i^0$ ) داشته باشد که پس از اعمال تابع تجمیع‌کننده Min در لایه اول و اعمال تابع تجمیع‌کننده Sum در لایه دوم، مقدار Embedding گره‌ها در لایه دوم با مقادیر اولیه یکی شوند (یعنی  $h_v^0 = h_v^2$  برای  $\forall v \in V$ ).

پاسخ:

$$H^{(L+1)} = D^{-1}AH^{(L)} \quad (\bar{\Gamma})$$

(ب) مقدار ورودی  $(i, j)$  در ماتریس انتقال به صورت زیر است:

- $\frac{1}{|D_{i,i}|}$  برای دو گره متصل  $i$  و  $j$
- 0 در غیر این صورت،

که به صورت  $D^{-1}A$  نوشته می‌شود.

نتیجه می‌گیریم مقدار Embedding گره  $i$  در لایه بعدی GNN برابر است با امید ریاضی Embedding گره  $i$  در یک مرحله حرکت تصادفی یکنواخت با شروع از گره  $i$  در گراف  $G$ .

(ج) مقدار ورودی  $(i, j)$  در ماتریس انتقال به صورت زیر است:

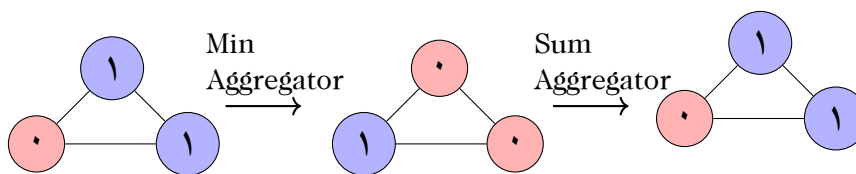
- $\frac{1}{3}$  هنگامی که  $i = j$
- $\frac{2}{3 \cdot |D_{i,i}|}$  برای دو گره متصل  $i$  و  $j$
- 0 در غیر این صورت،

که به صورت  $\frac{2}{3}D^{-1}A + \frac{1}{3}I$  نوشته می‌شود.  
با توجه به نتیجه بخش ب می‌توان رابطه ماتریسی را بصورت زیر نوشت:

$$H^{(L+1)} = \left(\frac{2}{3}D^{-1}A + \frac{1}{3}I\right)H^{(L)}$$

(د)

Initial Embeddings      Layer 1 Embeddings      Layer 2 Embeddings



## سوال چهارم: چالش‌های GNN ها (۸ نمره)

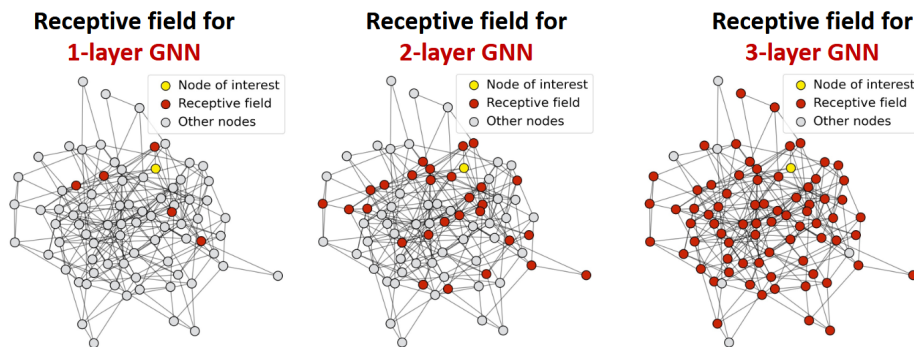
میدانیم یک لایه از شبکه GNN استاندارد از تابع زیر برای Aggregation و اعمال Activation Function استفاده می‌کند تا مقادیر گره‌ها را بروزرسانی کند.

$$\mathbf{h}_v^{(l)} = \sigma \left( \sum_{u \in \mathcal{N}(v)} \mathbf{w}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|\mathcal{N}(v)|} \right)$$

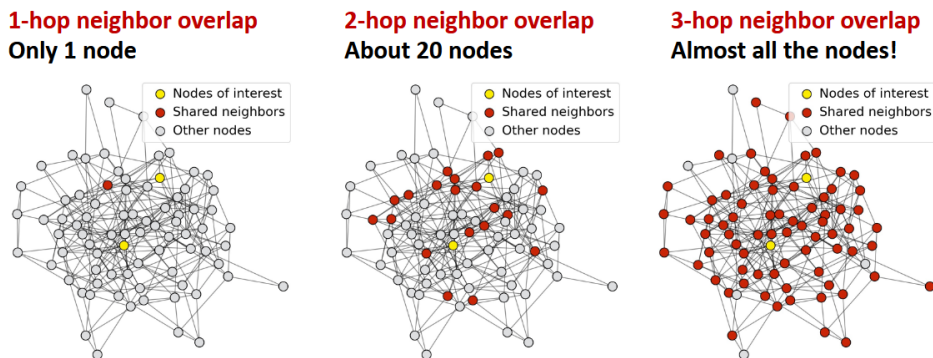
- (آ) یکی از چالش‌های اصلی GNN مشکل Over-smoothing است. این مشکل را توضیح دهید. دلیل بوجود آمدن این مشکل چیست؟ بدلیل چه خاصیتی شبکه‌های گرافی این چالش را دارند؟ استدلال خود را برای دو گره که در فاصله‌ی دوری از هم در گراف قرار دارند، با مثالی ترسیمی بیان کنید.
- (ب) روشی ارائه کنید که اگر نیاز به تعداد لایه‌های زیادی از GNN داریم، بتواند از بروز این مشکل جلوگیری کند. تابع آپدیت متناظر با روشی که بیان کردید را بنویسید.

پاسخ:

(آ) این مشکل بیان می‌کند که در شبکه‌های گرافی، با افزایش تعداد لایه‌های GNN مقدار Embedding گره‌ها به یک مقدار یکسان همگرا می‌شود. دلیل بوجود آمدن این موضوع هم افزایش تعداد لایه‌های GNN است.



شکل ۲: خاصیت اصلی شبکه گرافی که در مشکل Over-smoothing دخیل است receptive field بالای گره‌های این شبکه در تعداد لایه‌های بالاست.



شکل ۳: این مثال مشکل را ترسیم می‌کند که دو گره با فاصله نسبتاً زیاد چرا به مقدار یکسانی بعد از گذر از تعداد لایه‌های زیاد می‌رسند. افزایش همسایه‌های shared آن دو گره در لایه‌های آخر دلیل این موضوع است.

(ب) استفاده از ایده Skip connection و داشتن بلاک‌های residual که از مقدار Embedding لایه قبلی خود گره نیز استفاده می‌کند، کمک به جلوگیری از این مشکل خواهد کرد. تابع آپدیت به شکل زیر تغییر خواهد کرد:

$$h_v^{(l)} = \sigma \left( \sum_{u \in \mathcal{N}(v)} w^{(l)} \frac{h_u^{(l-1)}}{|\mathcal{N}(v)|} + h_v^{(l-1)} \right)$$

## سوال پنجم: Image Registration (۱۰ نمره)

با توجه به مطالبی که در کلاس آموختید، برای اینکه بتوانیم یک moving image را با fixed image انطباق دهیم و سپس مدلی داشته باشیم که بتوانیم فرآیند image registration را انجام دهیم:

(آ) یکی از روش‌ها برای این فرآیند استفاده از VoxelMorph است. ابتدا توضیح دهید ساختار VoxelMorph

به چه شکل است و با توجه به اینکه این روش Unsupervised است، چه مزیتی نسبت به روش Supervised که می‌توان برای آن استفاده کرد دارد؟

(ب) همان‌طور که می‌دانید، عبارت loss در **VoxelMorph** به این صورت تعریف می‌شود:

$$\mathcal{L} = \|m \circ \phi - f\| + \lambda \text{Reg}(\phi)$$

که در آن بخش اول مربوط به تطبیق تصاویر (images match) و بخش دوم تضمین‌کننده یک میدان هموار (smooth field) است.

مهم‌ترین مشکلی که در این loss وجود دارد چیست؟ و با چه روشی می‌توان این مشکل را حل کرد؟ (راهنمایی: برای به دست آوردن اطلاعات بیشتر، به مقاله **VoxelMorph** مراجعه کنید.)

پاسخ:

(آ) در **VoxelMorph** دو تصویر fixed image و moving image به عنوان ورودی دریافت می‌شوند و سپس از طریق یک شبکه عصبی CNN به deformation می‌رسیم. این فرآیند به صورت unsupervised انجام می‌شود.

یک روش دیگر نیز وجود دارد که در آن از یک فرد خواسته می‌شود تمامی train dataset را بررسی کرده و deformation را برای هر pair به دست آورد. در این حالت صرفاً به یک شبکه عصبی CNN نیاز داریم تا آن deformation را یاد بگیرد.

همان‌طور که مشخص است، روش supervised به علت زمان زیادی که از منابع انسانی می‌گیرد و همچنین به دلیل احتمال وجود خطا توسط فردی که deformation را به دست می‌آورد، توصیه نمی‌شود.

(ب) اصلی‌ترین مشکلی که این loss دارد، مشتق‌ناپذیر بودن آن است، زیرا هنگامی که می‌خواهیم backpropagation انجام دهیم، ماتریس متناظر آن از یک سری indicator function تشکیل می‌شود که مشتق‌ناپذیر است.

برای حل این مشکل، یک متد دیگر معرفی شده است که از آن می‌توان مشتق گرفت. که به صورت زیر می‌باشد:

$$m \circ \phi(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{Z}(\mathbf{p}')} m(\mathbf{q}) \prod_{d \in \{x, y, z\}} (1 - |p'_d - q_d|)$$

که در آن:

- $\mathcal{Z}(\mathbf{p}')$  مجموعه همسایگی‌های voxel  $\mathbf{p}'$  است.
- $d$  به ابعاد  $x, y, z$  اشاره دارد.

به این ترتیب، چون این روش از linear interpolation استفاده می‌کند، امکان محاسبه گرادیان‌ها و در نتیجه انجام back propagation طی بهینه‌سازی وجود دارد.

## سوال ششم (۱۰ نمره)

همان‌طور که از قبل می‌دانید یکی از روش‌های registration در تصاویر پزشکی **VoxelMorph** است. با توجه به سناریوی زیر به سوالات زیر پاسخ دهید:



- تعداد کمی داده‌ی T<sub>1</sub>-weighted داریم اما تصاویری که قرار است با آن‌ها کار کنیم ممکن است از contrast‌های مختلفی باشد (مثلاً تصاویر T<sub>2</sub>).

(الف) به نظر شما استفاده از **VoxelMorph** برای چنین سناریویی مناسب است؟ چرا؟

(ب) یکی دیگر از روش‌های register کردن تصاویر پزشکی **SynthMorph** است. برای سناریوی ذکرشده توضیح دهید استفاده از این روش مناسب است یا خیر؟ علت خود را نیز ذکر کنید. (برای آشنایی به روش - Synth Morph می‌توانید به [این مقاله](#) مراجعه کنید)

پاسخ:

(الف) استفاده از **VoxelMorph** برای چنین سناریویی مناسب نیست. این روش بر پایه registration model unsupervised learning-based است که نیاز به داده‌های contrast-یکسان برای آموزش دارد. در این روش فرض بر این است که داده‌های train و داده‌هایی که قرار است ثبت شوند از نظر ویژگی‌های شدت و contrast به یکدیگر شباهت دارند. در این سناریو که تصاویر contrast‌های مختلفی دارند (مثلاً T<sub>2</sub> در مقابل T<sub>1</sub>)، مدل نمی‌تواند به خوبی تعمیم یابد زیرا تفاوت‌های ساختاری ناشی از contrast باعث اختلال در عملکرد ثبت می‌شود. همچنین، تعداد کم داده‌های T<sub>1</sub>-weighted نیز محدودیت دیگر این روش است که وقتی دیتای کمتری داریم عموماً نتیجه‌ی مطلوبی را VoxelMorph به ما نمی‌دهد

(ب) **SynthMorph** برای این سناریو گزینه بسیار مناسبی است. این روش به جای تکیه بر داده‌های واقعی، از داده‌های مصنوعی که از نظر contrast، شدت، و نویز متنوع هستند برای آموزش استفاده می‌کند. در نتیجه، مدل نهایی توانایی تعمیم‌دهی به داده‌های واقعی با contrast‌های مختلف (مانند T<sub>2</sub>) را خواهد داشت. به طور خاص، SynthMorph از یک فرآیند داده‌افزایی استفاده می‌کند که طی آن تصاویر مصنوعی با contrast‌ها و تغییرات متنوع تولید می‌شوند. این امر به مدل اجازه می‌دهد تا به تفاوت‌های ناشی از contrast حساس نباشد و عملکرد قابل اعتمادی ارائه دهد. همچنین، این روش به دلیل عدم نیاز به داده‌های واقعی زیاد، محدودیت تعداد کم داده‌های T<sub>1</sub>-weighted را برطرف می‌کند. علاوه بر این، آزمایش‌ها نشان داده‌اند که SynthMorph می‌تواند در محیط‌هایی که تغییرات زیادی در داده‌های ورودی وجود دارد (مانند این سناریو) عملکرد بسیار خوبی داشته باشد.