به نام خدا

تمرین سری چهارم
درس تحلیل هوشمند تصاویر زیست پزشکی
دکتر محمد حسین رهبان

فرزان رحمانی
۴۰۳۲۱۰۷۲۵

سوال اول

(الف) چرا permutation invariance برای توابع aggregation در مسائل MIL ضروری است؟

در Multiple Instance Learning (MIL)، هدف این است که پیش‌بینی‌هایی بر اساس *bag* از نمونه‌ها (یا patches) انجام دهیم، جایی که برچسب bag به نمونه‌های درون آن بستگی دارد. در واقع انگار که یک Set از patch ها را داریم که در این مجموعه ترتیب اهمیتی ندارد. با این حال، ترتیب نمونه ها در bag به پیش بینی بی ربط است. برای مثال، مجموعه‌ای از تکه‌های تصویری که یک ناحیه تومور را نشان می‌دهند، بدون توجه به ترتیب patches، یکسان باقی می‌مانند.

یک تابع aggregation در MIL باید دارای خاصیت permutation invariance باشد تا اطمینان حاصل شود که پیش‌بینی‌های مدل بدون توجه به ترتیب نمونه‌ها در bag سازگار هستند. بدون permutation invariance، مدل می‌تواند پیش‌بینی‌های متفاوتی را برای یک bag مشابه با ترتیب مختلف تولید کند، که برای وظایف MIL معنی‌دار یا مطلوب نیست. بنابراین، توابع aggregation مورد استفاده در MIL باید ویژگی های نمونه ها را به گونه ای ترکیب کنند که مستقل از ترتیب آنها باشد.

مثال دقیق تر یک تصویر histopathology:

۱. در MIL، ما با bag هایی (مجموعه هایی: sets) از نمونه‌ها سر و کار داریم که ترتیب نمونه‌ها اطلاعات معنی‌داری را در خود ندارند. فرض کنید در حال تجزیه و تحلیل patch هایی از یک تصویر هیستوپاتولوژی برای تشخیص سرطان هستیم:

$$f(x_1, x_2, \dots, x_n) = f\left(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}\right)$$

که در آن π هر جایگشتی از index ها است.

۲. این که ابتدا به patch شماره ۱ نگاه کنیم یا patch شماره ۵، نباید بر پیش بینی نهایی ما در مورد اینکه آیا بافت دارای سلول های سرطانی است یا خیر تأثیر بگذارد. وجود سرطان بر اساس محتوای patch ها تعیین می شود، نه ترتیب آنها.
۳. ویژگی permutation invariance، پیش‌بینی‌های ثابتی را بدون توجه به نحوه ترتیب‌بندی نمونه‌های موجود در یک bag تضمین می‌کند و مدل را قوی‌تر و قابل اعتمادتر می‌کند.

(ب) مناسب بودن توابع Aggregation خاص در MIL

آ) تابع Maximum:

- تعریف: $f_{\max}(x_1, x_2, \dots, x_n) = \max(x_1, x_2, \dots, x_n)$
- Permutation Invariance: بله، تابع permutation invariant است زیرا حداکثر مقدار بدون توجه به ترتیب نمونه ها ثابت می ماند. (حداکثر مقدار یک مجموعه از اعداد به ترتیب آنها بستگی ندارد. $\max(x_1, x_2) = \max(x_2, x_1)$)

- مناسب بودن: این تابع برای مسائل MIL که برچسب bag با مهم ترین نمونه تعیین می شود (فرض اینکه یک bag مثبت است اگر حداقل یک instance مثبت باشد) مناسب است. به عنوان مثال، در تصویربرداری پزشکی، اگر وجود یک ویژگی خاص در هر نمونه ای نشان دهنده تشخیص مثبت باشد، تابع حداکثر با تمرکز بر شاخص ترین نمونه به طور موثر این موضوع را ثبت می کند. (به خوبی با مسائلی که وجود یک نمونه مهم (به عنوان مثال، یک patch تومور در تصویربرداری پزشکی) برچسب bag را تعیین می کند، هماهنگ است.)

ب) تابع Product:

- تعریف: $f_{\text{prod}}(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} x_i$
- Permutation Invariance: بله، تابع ضرب permutation invariant است زیرا حاصل ضرب مقادیر مستقل از ترتیب آنهاست. (حاصل ضرب مجموعه ای از اعداد مستقل از ترتیب آنهاست. $x_1 \times x_2 = x_2 \times x_1$)
- مناسب بودن: در حالی که permutation invariant است، تابع ضرب می تواند در عمل مشکل ساز باشد (Not generally suitable). به مقیاس مقادیر بسیار حساس است و ممکن است بی ثباتی عددی (numerical instability) را تقویت کند. اگر هر نمونه ای مقدار صفر داشته باشد، کل ضرب صفر می شود و به طور بالقوه سهم سایر نمونه ها را باطل می کند. علاوه بر این، برای تعداد زیادی از نمونه ها، محصول می تواند به صورت تصاعدی رشد یا کاهش یابد (grow or diminish exponentially) که منجر به بی ثباتی عددی (numerical instability) می شود. بنابراین، به طور کلی برای مسائل MIL مناسب نیست. علاوه بر این، با مفروضات معمول در وظایف MIL، که در آن سهم نمونه‌ها به‌جای multiplicative بودن، به طور کلی additive یا تحت سلطه نمونه‌های critical است، مطابقت ندارد.
- حالت خاص: البته به دلیل داشتن ویژگی permutation invariant می تواند در مسائل خاصی که همه instance ها (patches) در پیش بینی نهایی مشارکت دارند، کمک کند (with caveats). ولی به طور کلی مناسب مسائل MIL مانند تصویر هیستوپاتولوژی برای تشخیص سرطان نیست همان طور که در بالا توضیح دادیم.

ج) تابع Concatenation:

- تعریف: $f_{\text{concat}}(x_1, x_2, \dots, x_n) = [x_1, x_2, \dots, x_n]$
- Permutation Invariance: خیر، تابع concatenation یا الحاق permutation invariant نیست زیرا تغییر ترتیب نمونه ها نتیجه concatenation را تغییر می دهد. (ترتیب نمونه ها، بردار حاصل را تحت تأثیر قرار می دهد مثلا $[x_1, x_2] \neq [x_2, x_1]$)
- مناسب بودن: به دلیل حساسیت به ترتیب نمونه، تابع concatenation برای مسائل MIL نامناسب است. می‌تواند به نمایش‌های متفاوتی برای bag های یکسان ارائه‌شده با ترتیب‌های مختلف منجر شود، که منجر به یادگیری ناسازگار و تعمیم ضعیف می‌شود. در واقع، این تابع الزام اصلی permutation invariance در MIL را نقض می کند.
- محدودیت ها: بعد خروجی به تعداد نمونه ها بستگی دارد، که کار با bag های با اندازه های مختلف را دشوار می کند. لذا، نیاز به ملاحظات معماری اضافی (مانند padding) برای رسیدگی به مشکل bag های با اندازه متغیر دارد.

به طور خلاصه، برای توابع تجمع در MIL، permutation invariance برای اطمینان از نمایش bag های سازگار و قابل اعتماد ضروری (consistent and reliable bag representations) است. تابع ماکزیمم به طور کلی برای وظایف MIL مناسب است، به خصوص زمانی که برچسب bag به مهم ترین نمونه بستگی دارد. در مقابل، توابع ضرب و concatenation به ترتیب به دلیل بی ثباتی عددی (potential numerical issues) و حساسیت به ترتیب نمونه نامناسب هستند.

همچنین، سایر توابع permutation invariant مانند مکانیسم‌های mean pooling یا learned attention نیز معمولاً در کاربردهای MIL جدید تر استفاده می‌شوند، به ویژه attention زمانی که رابطه بین نمونه‌ها پیچیده‌تر از فرض MIL استاندارد باشد.

سوال دوم

(آ)

طبق فرمول داده شده، احتمال تعلق یک گره i به کلاس c به صورت زیر داده می شود:

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

که:

- $|N_i|$: تعداد همسایگان گره i
- $W(i,j)$: وزن یال بین گره‌های i و j (برای همه یال‌ها ۱ داده شده است)
- $P\big(Y_j = c\big)$: احتمال همسایه j متعلق به کلاس c

شرایط اولیه

- گره های دارای برچسب:

$$P(Y_3 = A) = 1, P(Y_3 = B) = 0$$

$$P(Y_5 = A) = 1, P(Y_5 = B) = 0$$

$$P(Y_8 = B) = 1, P(Y_8 = A) = 0$$

$$P(Y_{10} = B) = 1, P(Y_{10} = A) = 0$$

- گره های بدون برچسب:

$$for\ \ i \in \{1,2,4,6,7,9\}$$
$$P(Y_i = A) = P(Y_i = B) = 0.5 \quad (unbiased \quad initialization)$$

توجه داریم که مراحل را فقط باید برای گره های بدون برچسب انجام دهیم چرا که گره های دارای برچسب از قبل مشخص هستند.

راه حل ادامه سوال در صفحات بعد آمده است.

۲-(آ)

Subject: Year: Month: Date:

چون دو کلاس داریم پس طبق اصل متمم برای احتمال داریم:

$$P(Y_i=B) = 1 - P(Y_i=A)$$

$$P(Y_i=A) + P(Y_i=B) = 1$$

مرحله (تکرار) اول:

$$P(Y_1=A) = \frac{1}{2}(1\times\frac{1}{2}+1\times 1) = 0,75 \rightarrow P(Y_1=B) = 0,25$$
$$P(Y_2=A) = \frac{1}{3}(1\times 0,75+1\times 1+1\times\frac{1}{2}) = 0,75$$
$$P(Y_3=A) = \frac{1}{3}(0,75+0,5+0) \simeq 0,42$$
$$P(Y_4=A) = \frac{1}{4}(1+1+\frac{1}{2}+0) = 0,625$$
$$P(Y_5=A) = \frac{1}{2}(0,42+0) = 0,21$$
$$P(Y_6=A) = \frac{1}{4}(1+0,625+0+0) = 0,40625 \simeq 0,41$$

مرحله (تکرار) دوم:

$$P(Y_1=A) = \frac{1}{2}(0,75+1) = 0,875$$
$$P(Y_2=A) = \frac{1}{3}(0,875+1+0,42) = 0,765$$
$$P(Y_3=A) = \frac{1}{3}(0,765+0,21+0) = 0,325$$
$$P(Y_4=A) = \frac{1}{4}(1+1+0,40625+0) \simeq 0,402$$
$$P(Y_5=A) = \frac{1}{2}(0,325+0) = 0,1625 \simeq 0,16$$
$$P(Y_6=A) = \frac{1}{4}(1+0,402+0+0) = 0,4005 \simeq 0,4$$

بنابراین با توجه به اصل متمم داریم. $i=2,4,6$

$$P(Y_2=B) = 1-0,765 = 0,235,\quad P(Y_3=B) = 1-0,325 = 0,675$$

$$P(Y_4=B) = 1-0,402 = 0,398$$

$P(Y_2=B) = 0,235$
$P(Y_3=B) = 0,675$
$P(Y_4=B) = 0,398$

SOBHAN

(ب)

Subject: Year: Month: Date:

دو مرحله دیگر مرحله های update را تکرار می کنیم تا ببینیم به چه اعداد همگرا خواهیم شد.

مرحله (تکرار) سوم:

$P(Y_1=A) = \frac{1}{2}(0,745+1) = 0,8725$

$P(Y_2=A) = \frac{1}{3}(0,8725+1+0,325) = 0,7358\overline{3} \simeq 0,736$

$P(Y_4=A) = \frac{1}{3}(0,736+0,1425+0) = 0,2995$

$P(Y_6=A) = \frac{1}{4}(1+1+0,4+0) = 0,6$

$P(Y_7=A) = \frac{1}{2}(0,2995+0) = 0,14975$

$P(Y_9=A) = \frac{1}{4}(1+0,6+0+0) = 0,4$

مرحله (تکرار) چهارم:

$P(Y_1=A) = \frac{1}{2}(0,736+1) = 0,868 \simeq 0,87 > 0,5 \rightarrow$ کلاس A

$P(Y_2=A) = \frac{1}{3}(0,868+1+0,2995) = 0,7225 \simeq 0,72 > 0,5 \rightarrow$ کلاس A

$P(Y_4=A) = \frac{1}{3}(0,7225+0,14975+0) = 0,29075 \simeq 0,29 < 0,5 \rightarrow$ کلاس B

$P(Y_6=A) = \frac{1}{4}(1+1+0,4+0) = 0,6 > 0,5 \rightarrow$ کلاس A

$P(Y_7=A) = \frac{1}{2}(0,29075+0) = 0,145375 \simeq 0,15 < 0,5 \rightarrow$ کلاس B

$P(Y_9=A) = \frac{1}{4}(1+0,6+0+0) = 0,4 < 0,5 \rightarrow$ کلاس B

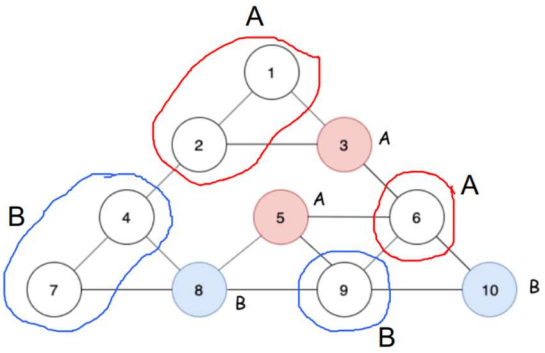
همان طور که در بالای ببینیم مقادیر احتمالات گره تقریباً همگرا شده اند و نسبت به مرحله قبل تغییرات در حدود ۰٫۰۱ است و بنابراین از این به بعد نسبت به آستانه ۰٫۵ تغییر قابل توجه نخواهند کرد پس می توانیم بگوییم گره ها در نهایت به کدام کلاس تعلق خواهند داشت.

for $i \in \{1,2,6\}$ $P(Y_i=A) > 0.5$ ← گره های 1 و 2 و 6 : کلاس A

for $i \in \{4,7,9\}$, $P(Y_i=A) < 0.5$ ← گره های 4 و 7 و 9 : کلاس B

SOBHAN

برچسب نهایی گره ها در این شکل نیز مشخص اند که رنگ قرمز نشانگر کلاس A و رنگ آبی نشانگر کلاس B است.



سوال سوم

(آ) رابطه ماتریسی برای Mean Aggregator:

معادله به روز رسانی تعبیه (embedding) برای گره i به صورت زیر بیان می شود:

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

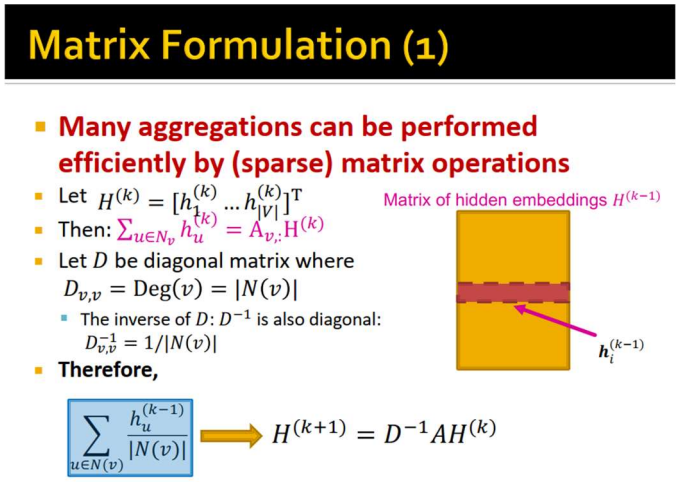
با استفاده از ماتریس قطری درجه D و ماتریس مجاورت A، می توان این معادله را برای همه گره های گراف به صورت زیر تعمیم داد:

$$H^{(l+1)} = D^{-1} A H^{(l)}$$

که:

- $H^{(l)}$ ماتریس تعبیه‌ها (embeddings) برای همه گره‌ها در لایه l است.
- A ماتریس مجاورت است. (represents connections)
- D^{-1} وارون ماتریس درجه گراف است که یک ماتریس قطری یا مورب است که در آن $D^{-1}{}_{ii} = \frac{1}{|N_i|}$ معکوس درجه گره i را نشان می دهد.
- $D^{-1}A$ ضریب میانگین نرمال شده برای همسایگان هر گره را محاسبه می کند مثلا اگر 5 همسایه داشته باشیم برای هر همسایه ضریب $\frac{1}{5}$ در نظر میگیرد تا میانگین گیری انجام شود در حین جمع در ضرب ماتریسی.

توضیحات بیشتر در این اسلاید آمده است:



(ب) ماتریس انتقال (Transition Matrix) برای Random Walk:

برای یک random walk بر روی گراف G، ماتریس انتقال T به گونه ای تعریف می شود که:

$$T[i,j] = P(i \to j)$$

جایی که $P(i \to j)$ احتمال حرکت از گره i به j است. اگر i و j به هم متصل باشند، این مقدار برابر با $\frac{1}{\deg(i)}$ است و اگر متصل نباشند برابر با 0 است. در واقع با توجه به اینکه A یک ماتریس صفر و یکی است میتوانیم احتمال حرکت از گره i به j به صورت $P(i \to j) = \frac{A_{i,j}}{\deg(i)}$ بنویسم. پس، با استفاده از ماتریس مجاورت A و ماتریس درجه D، ماتریس انتقال T به صورت زیر است:

$$T = D^{-1}A$$

رابطه بین (الف) و (ب):

ماتریس $D^{-1}A$ در هر دو بخش یکسان است. این نشان می دهد که تابع mean aggregator در GNN ها معادل انجام یک مرحله از یک random walk روی گراف است (a single step of a random walk on the graph). بنابراین، فرآیند به روز رسانی جاسازی (embedding) در GNN ها را می توان به عنوان انتشار اطلاعات از طریق random walk تفسیر کرد که در آن جاسازی جدید هر گره میانگین جاسازی های همسایگانش (میانگین اطلاعات منتشر شده از طریق random walk) است.

(ج) رابطه ماتریسی برای Aggregator با Skip Connection:

معادله جدید به روز رسانی embedding به صورت زیر است:

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

پس در شکل ماتریسی، این می شود:

$$H^{(l+1)} = \frac{1}{3}H^{(l)} + \frac{2}{3}D^{-1}AH^{(l)}$$

میتوانیم از $H^{(l)}$ فاکتور بگیریم:

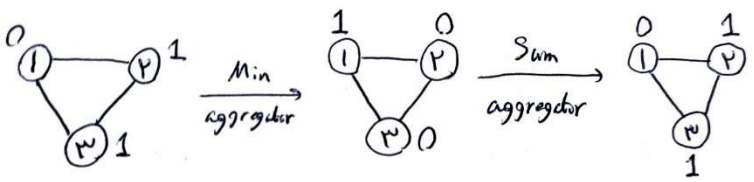
$$H^{(l+1)} = \left(\frac{1}{3}I + \frac{2}{3}D^{-1}A\right)H^{(l)}$$

جایی که I ماتریس هویت (identity matrix) است. این عبارت نشان می دهد که skip connection یک self-loop influence را به فرآیند aggregation اضافه می کند.

همچنین به جای $\frac{1}{3}$ و $\frac{2}{3}$ میتوانیم به صورت پارامتری رابطه بالا را جایگزین کنیم تا با تغییر α بتوانیم اهمیت skip connection را کم یا زیاد کنیم.

$$H^{(l+1)} = (\alpha I + (1-\alpha)D^{-1}A)H^{(l)}$$

(د) یافتن یک گراف برای شرایط خواسته شده: (توضیحات مفصل در صفحه بعد)



ما میخواهیم یک گراف $G(V,E)$ بسازیم که:

۱. $|V| \geq 3$
۲. $\forall v \in V, \deg(v) \geq 2$
۳. پس از اعمال Min Aggregator در لایه اول و Sum Aggregator در لایه دوم، embedding ها به مقادیر اولیه خود باز می گردند.

ساخت گراف:

فرض کنید $G(V,E)$ مثلثی باشد با ۳ گره ($V = \{1,2,3\}$) و یال هایی که یک cycle را تشکیل می دهند: $E = \{(1,2),(2,3),(3,1)\}$.

تعبیه های اولیه (Initial Embeddings):

فرض کنید تعبیه‌های اولیه به صورت زیر باشد:

$$h_1^{(0)} = 0, \quad h_2^{(0)} = 1, \quad h_3^{(0)} = 1$$

یعنی:

$$H^{(0)} = [0, \quad 1, \quad 1]^T$$

لایه اول (Min Aggregator):

با استفاده از Min aggregator، تعبیه هر گره به عنوان حداقل همسایگانش به روز می شود:

$$h_1^{(1)} = \min\left(h_2^{(0)}, h_3^{(0)}\right) = \min(1,1) = 1$$

$$h_2^{(1)} = \min\left(h_1^{(0)}, h_3^{(0)}\right) = \min(0,1) = 0$$

$$h_3^{(1)} = \min\left(h_1^{(0)}, h_2^{(0)}\right) = \min(0,1) = 0$$

بنابراین:

$$H^{(1)} = [1, \quad 0, \quad 0]^T$$

لایه دوم (Sum Aggregator):

با استفاده از Sum Aggregator، جاسازی هر گره به عنوان مجموع همسایگان آن به روز می شود:

$$h_1^{(2)} = h_2^{(1)} + h_3^{(1)} = 0 + 0 = 0$$

$$h_2^{(2)} = h_1^{(1)} + h_3^{(1)} = 1 + 0 = 1$$

$$h_3^{(2)} = h_1^{(1)} + h_2^{(1)} = 1 + 0 = 1$$

بدین ترتیب:

$$H^{(2)} = [0, \quad 1, \quad 1]^T$$

همان طور که میبینیم مقدار تعبیه ها در لایه دوم با مقدار اولیه یکی شد و تمامی شروط خواسته شده نیز ارضا شدند.

$$H^{(2)} = H^{(0)} = [0, \quad 1, \quad 1]^T$$

پس گراف نهایی یافته شده این گراف است:

$$V = \{1, \quad 2, \quad 3\}, \qquad E = \{(1,2), \quad (2,3), \quad (3,1)\},$$
$$h_1^{(0)} = 0, \quad h_2^{(0)} = 1, \quad h_3^{(0)} = 1, \qquad H^{(0)} = [0, \quad 1, \quad 1]^T$$

که همه شرایط را برآورده می کند.

سوال چهارم

(آ) مشکل Over-Smoothing در GNN:

Over-Smoothing چیست؟

هموارسازی بیش از حد (Over-Smoothing) زمانی اتفاق می‌افتد که تعبیه‌های (embeddings) گره (node) به طور فزاینده‌ای شبیه (یا حتی یکسان) می‌شوند، چرا که ما وقتی لایه‌های GNN بیشتری را روی هم قرار می‌دهیم باعث می‌شود که مدل قدرت تمایز را از دست بدهد. اساساً، همه گره ها به نمایش های یکسان یا بسیار مشابه همگرا می شوند. در واقع، هموارسازی بیش از حد در شبکه‌های عصبی گراف (GNN) زمانی اتفاق می‌افتد که پس از انباشتن چندین لایه، embedding های گره $h_v^{(l)}$ همه گره‌ها در گراف تقریباً یکسان (یا غیرقابل تمایز یا indistinguishable) می‌شوند. این منجر به از دست رفتن اطلاعات در سطح گره می شود و تمایز گره ها را بر اساس embedding آنها دشوار می کند. در واقع انگاری که embedding همه گره به یک بردار همگرا می شوند.

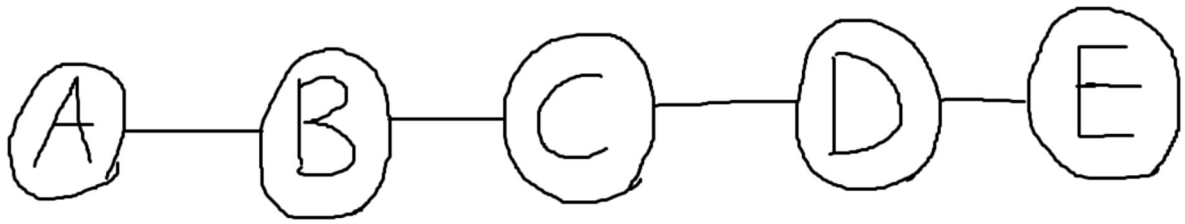
دلیل Over-Smoothing:

علت اصلی Over-Smoothing، تجمع مکرر اطلاعات از گره های همسایه در چندین لایه (aggregation of information from neighboring nodes across multiple layers) است. با هر لایه، جاسازی (embedding) گره ها اطلاعات بیشتری را از همسایگان خود ترکیب می کنند و با افزایش تعداد لایه ها، جاسازی گره ها در سراسر گراف تمایل به همگرایی دارند. این به این دلیل است که مکانیسم aggregation مانند یک فرآیند انتشار (diffusion process) عمل می کند، جایی که اطلاعات به طور یکنواخت در سراسر گراف پخش می شود. در واقع اگر با در نظر گرفتن مدل message-passing به گراف نگاه کنیم با افزایش تعداد لایه ها همه گره ها از تمام گره های دیگر message دریافت میکنند و پس از aggregate شدن این message ها تمام گره ها به جاسازی (embedding) مشابه می رسند.

چرا GNN ها این چالش را دارند:

۱. مکانیسم انتشار (Propagation Mechanism): GNN تعبیه‌های همسایه‌ها (neighbors' embeddings) را در هر لایه aggregate می‌کند، و هر چه گره از همسایه‌های خود دورتر باشد، تعبیه‌های آن‌ها بیشتر میانگین گرفته می‌شود و منحصربه‌فرد (uniqueness) بودن را از دست می‌دهد.
۲. Graph Connectivity: برای دو گره که در گراف از هم دور هستند، repeated aggregation یا همان تجمع مکرر (به عنوان مثال، از طریق $normalized\ summation - |\mathcal{N}(v)|$) باعث می شود جاسازی آنها شبیه به یکدیگر شود، به خصوص در گراف های densely connected یا گراف هایی که مسیرهای زیادی بین گره های آنها وجود دارد.

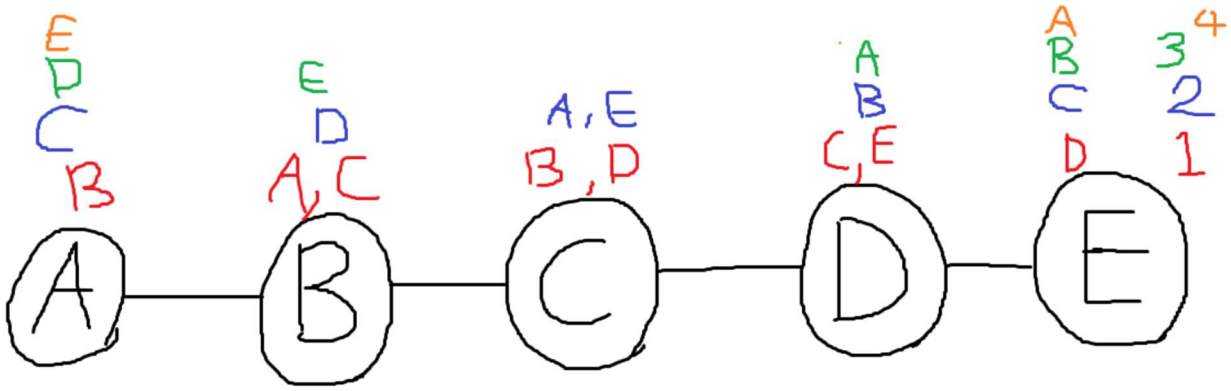
مثال با دو گره دور از هم در یک گراف دلخواه:



گراف بالا را در نظر بگیرید. در این گراف گره های A و E که دور ترین فاصله را از هم دیگر دارند تحلیل میکنیم. در ابتدا، گره های A و E دارای ویژگی های متمایز (distinct features or distinct embeddings) هستند. بعد از هر لایه:

- لایه ۱: به روز رسانی A و E بر اساس همسایگان مستقیم خود رخ می دهد. (به ترتیب B و D)
- لایه ۲: اطلاعات از C به هر دو می رسد یعنی در به روز رسانی A و E اطلاعات گره C تاثیر میگذارد.
- لایه ۳: اطلاعات از D به A می رسد و همچنین اطلاعات از B به E می رسد و با این اطلاعات به روز رسانی انجام می شود.
- لایه ۴: A شروع به نفوذ اطلاعات E می کند. E هم از A تأثیر می گیرد یعنی اطلاعات از E به A می رسد.
- بعد از چندین لایه: representation های A و E با وجود دور بودن از هم بسیار شبیه می شوند.

در ادامه شماتیک نشر اطلاعات را میبینید:



دلیل ریاضی این موضوع:

- با تجمع مکرر همسایگی:

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

- این عملیات اساساً میانگین گیری مکرر (repeated averaging) را انجام می دهد.
- وقتی که $l \to \infty$ ، جاسازی‌ها (embeddings) به یک توزیع ثابت (stationary distribution) همگرا می‌شوند (مشابه نحوه همگرا شدن random walks)
- این امر به ویژه برای گره هایی که باید بر اساس نقش یا ویژگی های ساختاری خود بازنمایی های متفاوتی (unique) داشته باشند مشکل ساز است.

(ب) روش هایی برای جلوگیری از Over-Smoothing:

یکی از روش های موثر برای کاهش Over-Smoothing، Residual Connections (یا Skip Connections) است. Residual Connections به هر گره اجازه می دهد تا تعبیه اولیه خود را از لایه های قبلی حفظ کند، در نتیجه اطلاعات مربوط به گره را حفظ می کند در حالی که هنوز اطلاعات همسایگان را aggregate می کند.

عملکرد به روز رسانی با Residual Connections:

embedding به روز شده گره v در لایه l را می توان به صورت زیر نوشت:

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

که:

- σ: تابع فعال سازی (به عنوان مثال، ReLU یا sigmoid)
- $\boldsymbol{W}^{(l)}$: ماتریس وزن قابل یادگیری در لایه l
- $\frac{h_u^{(l-1)}}{|\mathcal{N}(v)|}$: تجمع همسایه عادی شده.
- α : یک هایپر پارامتر است که سهم Residual Connections را کنترل می کند $(0 < \alpha < 1)$
- $h_v^{(l-1)}$: بازنمایی گره از لایه قبلی است.

در معادله بالا میتوانیم از یک ماتریس وزن هم مانند اسلاید های درس استفاده کنیم ولی در معادله بالا ما کنترل بیشتری رو α داریم و میتوانیم با افزایش آن تاثیر بازنمایی خود گره را بیشتر و همسایگان را کمتر کنیم:

Trainable weight matrices
(i.e., what we learn)

$$\mathrm{h}_v^{(0)} = \mathrm{x}_v$$
$$\mathrm{h}_v^{(k+1)} = \sigma\left(\mathrm{W}_k \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(k)}}{|\mathrm{N}(v)|} + \mathrm{B}_k \mathrm{h}_v^{(k)}\right), \forall k \in \{0..K-1\}$$
$$\mathrm{z}_v = \mathrm{h}_v^{(K)}$$

مزایای اتصالات باقیمانده (Residual Connections):

۱. حفظ هویت گره: original embedding $h_v^{(l-1)}$ به هر لایه اضافه می‌شود و از همگرا شدن سریع جاسازی‌های گره جلوگیری می‌کند.
۲. دستیابی به GNNهای عمیق‌تر اجازه می‌دهد: شبکه می‌تواند لایه‌های بیشتری را بدون مواجهه با over-smoothing شدید روی هم قرار دهد و امکان ثبت وابستگی‌های دوربرد در نمودار را فراهم کند.

روش های جایگزین برای جلوگیری از Over-Smoothing:

۱. Graph Attention Networks (GAT):

به‌جای میانگین‌گیری ساده، وزن‌های توجه قابل یادگیری را به همسایگان اختصاص دهیم تا مدل به طور انتخابی روی همسایگان مهم تمرکز کند.

تابع به روز رسانی شده:

$$h_v^{(l)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} \boldsymbol{W}^{(l)} h_u^{(l-1)}\right)$$

که α_{vu} وزن توجه همسایه u است و از فرمول زیر به دست می آید.

$$e_{uv} = \mathrm{LeakyReLU}(a^{\top} \cdot [Wh_u \,|\, Wh_v])$$

$$\alpha_{uv} = \frac{\exp(e_{uv})}{\sum_{k \in \mathcal{N}(v)} \exp(e_{vk})}$$

۲. DropEdge Regularization:

برای کاهش انتشار اطلاعات اضافی و افزایش تنوع در جاسازی ها (embeddings)، یال ها را به طور تصادفی در طول آموزش drop کنیم.

۳. استفاده از تکنیک های Regularization مانند این مقاله (https://arxiv.org/abs/1909.03211):

- روش: تنظیم کننده هایی (regularizers را معرفی کنیم که over-smoothing را جریمه می کنند.
- تابع ضرر به روز رسانی شده چنین میشود:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \cdot \text{MADGap}(H)$$

- اثر: تنوع در نمایش گره ها را تشویق می کند.

۴. Jumping Knowledge Networks:

- روش: خروجی‌های لایه‌های مختلف را ترکیب کنیم که به مدل اجازه می‌دهد به‌طور adaptively مرتبط‌ترین خروجی‌های لایه را برای هر گره انتخاب کند.
- به عنوان مثال اگر از concatenation استفاده کنیم تابع به روز رسانی شده چنین می شود:

$$h_v = \text{concat}\left(h_v^{(0)}, h_v^{(1)}, \ldots, h_v^{(L)}\right)$$

- اثر: اطلاعات چند مقیاسی (multi-scale information) را حفظ می‌کند و over-smoothing را کاهش می‌دهد.

۵. Layer Normalization:

Embedding ها را در هر لایه نرمال کنیم تا تعادل بین اطلاعات aggregated و retained حفظ شود.

عملکرد به روز رسانی:

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

سوال پنجم

(آ) ساختار VoxelMorph و مزایای آن نسبت به روش های Supervised

ساختار VoxelMorph:

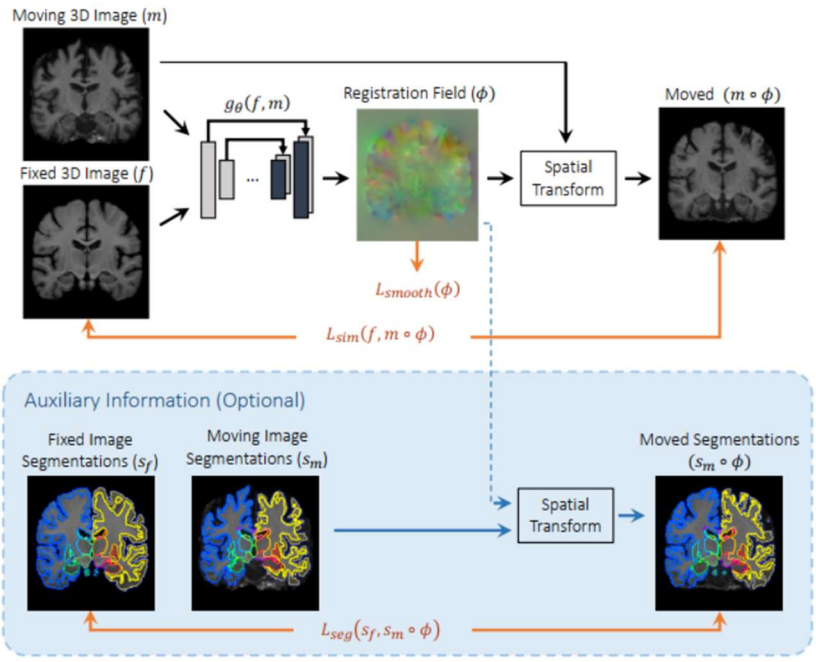


Fig. 2: Overview of the method. We learn parameters θ for a function $g_\theta(\cdot,\cdot)$, and register 3D volume m to a second, fixed volume f. During training, we warp m with ϕ using a spatial transformer function. Optionally, auxiliary information such as anatomical segmentations $\mathbf{s_f}, \mathbf{s_m}$ can be leveraged during training (blue box).

VoxelMorph یک framework مبتنی بر یادگیری عمیق است که برای ثبت تصاویر deformable بدون نظارت (unsupervised deformable image registration) طراحی شده است. معماری آن به طور کلی شامل:

۱. شبکه رمزگذار-رمزگشا (Encoder-Decoder Network): مدل یک جفت تصویر (fixed and moving) را به عنوان ورودی می گیرد و رمزگذار آنها را از طریق لایه های کانولوشنی پردازش می کند تا یک نمایش ویژگی با ابعاد پایین ایجاد کند. سپس رمزگشا ویژگی های کدگذاری شده را به وضوح اولیه upsample می کند و dense voxel-wise registration را امکان پذیر می کند. Skip Connection هل نیز به حفظ جزئیات فضایی در طول upsampling کمک می‌کند. این شبکه دارای معماری UNet است که جزییات آن در زیر آمد است.

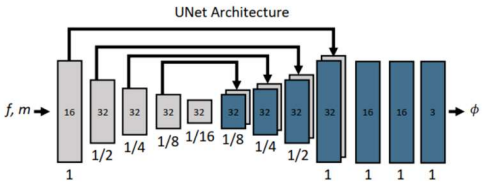


Fig. 3: Convolutional UNet architecture implementing $g_\theta(f, m)$. Each rectangle represents a 3D volume, generated from the preceding volume using a 3D convolutional network layer. The spatial resolution of each volume with respect to the input volume is printed underneath. In the decoder, we use several 32-filter convolutions, each followed by an upsampling layer, to bring the volume back to full resolution. Arrows represent skip connections, which concatenate encoder and decoder features. The full-resolution volume is further refined using several convolutions.

۲. Spatial Transform Network (Spatial Transformation Function): این ماژول deformation field یا همان میدان تغییر شکل (پیش بینی شده توسط شبکه UNet) را به تصویر moving اعمال می کند و آن را با تصویر fixed تراز می کند.
۳. Deformation Field Prediction: پس در نهایت مدل یک smooth deformation field به نام ϕ را پیش‌بینی می‌کند، که برای warp کردن تصویر متحرک (moving image) به تصویر ثابت (fixed image) استفاده می‌شود.
۴. Auxiliary Information (Optional): به صورت اختیاری برای بهبود عملکرد، اطلاعات کمکی مانند anatomical segmentationبا نماد s_m, s_f را می توان در طول آموزش استفاده کرد. (جعبه آبی در شکل بالا)

مزایای روش Unsupervised:

- بدون نیاز به برچسب‌های Ground Truth: برخلاف روش‌های نظارت شده (supervised)، VoxelMorph به ground truth deformation fields از پیش برچسب‌گذاری‌شده تکیه نمی‌کند، که می‌تواند چالش‌برانگیز و پرهزینه باشد. در واقع، تولید این ground truth ها می تواند کار labor-intensive باشد و ممکن است نوع deformation های آموخته شده را محدود کند.
- Scalability: روش‌های بدون نظارت مقیاس‌پذیرتر هستند، زیرا به داده‌های برچسب‌گذاری شده تکیه نمی‌کنند، که ممکن است کمیاب یا گران باشد.
- Generality یا قابل تطبیق در میان مجموعه‌های داده مختلف (Adaptable Across Datasets): طبیعت بدون نظارت، آن را با مجموعه داده‌های جدید بدون نیاز به حاشیه‌نویسی (re-annotation) مجدد سازگار می‌کند. به بیان دیگر، ماهیت بدون نظارت VoxelMorph به آن اجازه می دهد تا به مجموعه داده های و registration tasks مختلف تعمیم یابد، زیرا مستقیماً از شدت تصویر و الگوهای فضایی یاد می گیرد.
- Simpler Training Pipeline: نیاز به ایجاد یا تنظیم دستی داده های آموزشی paired را از بین می برد که فرآیند آموزش را ساده تر می کند.
- Flexibility: VoxelMorph می‌تواند اطلاعات کمکی (مانند segmentation maps) را در صورت وجود ترکیب کند، اما برای آموزش به آن نیازی ندارد.

(ب)

مهم ترین مشکل تابع ضرر در VoxelMorph، $\mathcal{L} = ||m \circ \phi - f|| + \lambda \mathrm{Reg}(\phi)$، trade-off بالقوه بین عبارت تطبیق یا شباهت تصویر ($||m \circ \phi - f||$) و عبارت منظم‌سازی ($\mathrm{Reg}(\phi)$) است. دو عبارت در تابع ضرر دارای اهداف رقابتی هستند. این trade-off می تواند منجر به موارد زیر شود:

۱. Over-smoothing of the deformation field: اگر λ، وزن منظم‌سازی، خیلی زیاد باشد، ممکن است عبارت منظم‌سازی غالب باشد و باعث شود که میدان تغییر شکل ϕ بیش از حد smooth باشد. این می تواند منجر به suboptimal alignment تصویر متحرک m با تصویر ثابت f شود، به ویژه در مناطقی که تفاوت های anatomical واضح دارند.
۲. Irregular or unrealistic deformations: اگر λ خیلی کم باشد، ممکن است عبارت شباهت غالب باشد و به مدل اجازه می‌دهد تا شدت‌های تصویر منطبق را به قیمت ایجاد میدان‌های تغییر شکل غیرقابل قبول یا نامنظم اولویت‌بندی کند.

یک راه حل شامل متعادل کردن دقیق دو عبارت در تابع ضرر است:

- Adaptive Regularization: تنظیم پویا وزن λ در طول آموزش بر اساس پیشرفت بهینه سازی می تواند به حفظ تعادل بین دقت تطبیق و نظم تغییر شکل (deformation regularity) کمک کند.
- Diffeomorphic Regularization: همانطور که در مقاله پیشنهاد شده است، اعمال محدودیت های دیفئومورفیک (تبدیل های حفظ توپولوژی) می تواند smoothness و invertibility میدان تغییر شکل را بهبود بخشد و از folding یا نگاشت غیرواقعی جلوگیری کند.
- آموزش Multiscale: ترکیب ویژگی‌های چند مقیاسی (different spatial scales) می‌تواند میدان تغییر شکل را با مدیریت هم‌ترازی درشت و ریزدانه اصلاح کند. از توابع ضرر multi-resolution برای متعادل کردن تراز global و local استفاده کنیم.

- Auxiliary Loss Functions: معرفی مؤلفه‌های ضرر اضافی، مانند segmentation-based Dice loss ، می‌تواند مدل را برای تراز کردن مناطق آناتومیکی خاص با دقت بیشتری راهنمایی کند. این اطلاعات کمکی روشی ساختاریافته برای بهبود registration بدون به خطر انداختن smoothness ارائه می دهد.

هدف این تکنیک ها بهبود trade-off و اطمینان از این است که تابع ضرر به طور موثر دقت هم ترازی را با قابل قبول بودن فیزیکی متعادل می کند.

به علاوه دو مشکل دیگر نیز در مقاله به آنها اشاره شده است که در زیر آمده اند:

B. Spatial Transformation Function

The proposed method learns optimal parameter values in part by minimizing differences between $m \circ \phi$ and f. In order to use standard gradient-based methods, we construct a differentiable operation based on spatial transformer networks [42] to compute $m \circ \phi$.

For each voxel $\mathbf{p}$, we compute a (subpixel) voxel location $\mathbf{p}' = \mathbf{p} + \mathbf{u}(\mathbf{p})$ in m. Because image values are only defined at integer locations, we linearly interpolate the values at the eight neighboring voxels:

$$m \circ \phi(\mathbf{p}) = \sum_{\mathbf{q} \in \mathcal{Z}(\mathbf{p}')} m(\mathbf{q}) \prod_{d \in \{x,y,z\}} (1 - |\mathbf{p}'_d - \mathbf{q}_d|), \qquad (3)$$

where $\mathcal{Z}(\mathbf{p}')$ are the voxel neighbors of $\mathbf{p}'$, and d iterates over dimensions of Ω. Because we can compute gradients or sub-gradients,[3] we can backpropagate errors during optimization.

به منظور استفاده از روش‌های مبتنی بر گرادیان استاندارد، یک عملیات مشتق پذیر بر اساس شبکه‌های spatial transformer برای محاسبه $m \circ \phi$ می‌سازیم. همچنین از آنجایی که مقادیر تصویر فقط در مکان های اعداد صحیح تعریف می شوند، ما مقادیر را در هشت وکسل مجاور به صورت خطی درون یابی می کنیم.

یکی دیگر از مشکلات هم مربوط به نواحی با non-positive Jacobian determinant است که ممکن است منجر به folding شود. برای این موارد، استفاده از Regularization که باعث محدود کردن دامنه تغییرات و جریمه کردن negative Jacobian determinant و تنظیم بهتر هایپرپارامتر هاست.

the number of voxels for which the determinant of the Jacobian is non-positive. Table IV provides the quantitative regularity measure for all γ values, showing that VoxelMorph deformation regularity degrades slowly as a function of γ (shown on a log scale), with roughly 0.2% of the voxels exhibiting folding at the lowest parameter value, and *at most* 2.3% when $\gamma = 0.1$. Deformations from models that don't encourage smoothness, at the extreme value of $\gamma = \infty$, exhibit 10–13% folding voxels. A lower γ value such as $\gamma = 0.01$ therefore provides a good compromise of high Dice scores for all structures while avoiding highly irregular deformation fields, and avoiding overfitting as described above. Fig 10 shows examples of deformation fields for $\gamma = 0.01$ and $\gamma = \infty$, and we include more figures in the supplemental material for each experimental setting.

4) Testing on Manual Segmentation Maps: We also test these models on the manual segmentations in the Buckner40

سوال ششم

(الف) خیر VoxelMorph مناسب نیست.

VoxelMorph یک روش ثبت تصویر بدون نظارت (unsupervised image registration) است که برای تراز کردن موثر تصاویر پزشکی طراحی شده است. با این حال، در سناریوی داده شده که در آن مقدار کمی داده T1-weighted وجود دارد و نیاز به کار با کنتراست های مختلف (به عنوان مثال، تصاویر T2-weighted) وجود دارد، VoxelMorph ممکن است با چالش هایی روبرو شود:

۱. وابستگی به contrast های مشابه (Dependence on Similar Contrasts): VoxelMorph روی تصاویر با کنتراست مشابه عملکرد خوبی دارد زیرا بر معیارهای تشابه مبتنی بر شدت (intensity-based similarity metrics) تصویر (مثلاً mean squared error یا normalized cross-correlation) متکی است. این معیارها زمانی که تصاویر دارای کنتراست های متفاوتی هستند، مانند T1 و T2 کمتر موثر هستند. به بیان دیگر، VoxelMorph برای dissimilarity loss بر تشابه شدت تصویر متکی است. اگر تصاویر moving و fixed دارای کنتراست های بسیار متفاوتی باشند (به عنوان مثال، T1 در مقابل T2)، متریک شباهت ممکن است شکست بخورد و منجر به registration نادرست شود.
۲. تعمیم پذیری محدود در میان Contrast ها: مدل هایی که با استفاده از VoxelMorph بر روی تصاویر T1-weighted آموزش دیده اند، ممکن است به خوبی به تصاویر T2-weighted تعمیم ندهند، مگر اینکه در طول آموزش به طور واضح در معرض آن تضادها قرار گیرند.
۳. داده های آموزشی محدود (Limited Training Data): VoxelMorph برای آموزش به داده های برچسب دار یا بدون برچسب کافی نیاز دارد. با مقدار کمی داده T1-weighted، توانایی آن برای تعمیم به کنتراست های نادیده (unseen contrasts) مانند تصاویر T2-weighted ممکن است به خطر بیفتد.

نتیجه گیری: استفاده از VoxelMorph در این سناریو ایده‌آل نیست مگر اینکه داده‌های کافی از تمام کنتراست‌ها (به عنوان مثال، T1 و T2) برای آموزش در دسترس باشد. فاقد ویژگی‌های ذاتی contrast-invariant است که برای این سناریو بسیار مهم است.

(ب) بله SynthMorph مناسب است.

SynthMorph یک روش contrast-invariant registration است که وابستگی به داده های تصویربرداری به دست آمده (acquired) را حذف می کند. طراحی آن، آن را به ویژه برای سناریوهایی مانند آنچه توضیح داده شد مناسب می کند:

۱. Contrast-Invariant Framework (Contrast-Invariant Features): SynthMorph بر روی تصاویر مصنوعی تولید شده از random label maps آموزش می بیند و شبکه را در معرض طیف گسترده ای از کنتراست ها و اشکال قرار می دهد. این استراتژی تضمین می‌کند که مدل می‌تواند بین contrast ها، از جمله موارد دیده نشده مانند T2، تعمیم یابد. به بیان دیگر، SynthMorph بر روی داده‌های مصنوعی تولید شده با کنتراست‌های مختلف آموزش داده شده است، و آن را قادر می‌سازد تا در زمان آزمون به unseen contrasts (مانند T2 یا سایر مودالیته‌ها) تعمیم دهد.
۲. عدم وابستگی به داده های اکتسابی (No Dependency on Acquired Data): برخلاف VoxelMorph، SynthMorph برای آموزش به داده های واقعی نیاز ندارد. این روش، داده‌های آموزشی را با تضادهای دلخواه ترکیب می‌کند، که وقتی تنها مقدار کمی از داده‌های T1 در دسترس باشد بسیار سودمند است. در واقع، از synthesized label maps برای ایجاد training pairs متنوع استفاده می‌کند که آن را برای روش‌های مختلف تصویربرداری قوی می‌کند.
۳. موثر در همه Modality ها: همانطور که در ارزیابی های عملکردش نشان داده شده است، SynthMorph به registration قوی برای سناریوهای cross-contrast (به عنوان مثال، T1-to-T2) بدون آموزش مجدد دست می یابد.

نتیجه گیری: SynthMorph برای سناریوی توصیف شده بسیار مناسب است. ماهیت contrast-agnostic آن عملکرد قوی را حتی با داده‌های محدود T1 تضمین می‌کند و آن را در مقایسه با VoxelMorph برای تسک های cross-contrast registration انتخابی برتر می‌سازد.

در ادامه بخش هایی از مقاله که به برتری SynthMorph بر VoxelMorph در تسک های Constrast Variant Registration اشاره می کنند آمده است:

A. Generalizability

A significant challenge in the deployment of neural networks is their generalizability to image types unseen during training. Existing learning methods like VoxelMorph achieve good registration performance but consistently fail for new MRI contrasts at test time. For example, vm-ncc is trained on T1w pairs and breaks down both across contrasts (e.g. T1w-T2w) *and* within new contrasts (e.g. T2w-T2w). The SynthMorph strategy addresses this weakness and makes networks resilient to contrast changes by exposing them to a wide range of synthetic images, far beyond the shapes and contrasts typical of MRI. This approach obviates the need for retraining to register images acquired with a new sequence.

Training conventional VoxelMorph with a loss evaluated on T1w images while augmenting the input contrasts enables the transfer of domain-specific specific knowledge to cross-contrast registration tasks. However, the associated decrease in within-contrast performance indicates the benefit of SynthMorph: learning to match anatomical features independent of their appearance in the gray-scale images.

The choice of optimum hyperparameters also is an important problem for many deep learning applications. While the

2) Results: Fig. 11 compares registration accuracy as a function of the moving-image MRI contrast for baseline methods and SynthMorph. In both the multi-FA and the multi-TI data, we obtain broadly comparable results for all methods when the moving and fixed image have T1w-like contrast. However, the performance of ANTs, NiftyReg and learning baselines decreases with increasing contrast differences, whereas SynthMorph remains largely unaffected.

Fig. 12 shows the variability of the response of each network layer to varying MRI contrast of the same anatomy (shown in Fig. 9). Compared to VoxelMorph, the feature variability within the deeper layers is significantly lower for the SynthMorph models. Fig. 10 illustrates this result, containing example feature maps extracted from the last network layer before the SVF is formed.

Overall, SynthMorph models exhibit substantially less variability in response to contrast changes than all other methods tested, indicating that the proposed strategy does indeed encourage contrast invariance.

پایان