

به نام خدا
دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه پایانی درس شبکه‌های پیچیده

استاد درس: دکتر مصطفی حقیر چهرقانی

دانشجو: فاطمه غلامزاده

۹۹۱۳۱۰۰۳

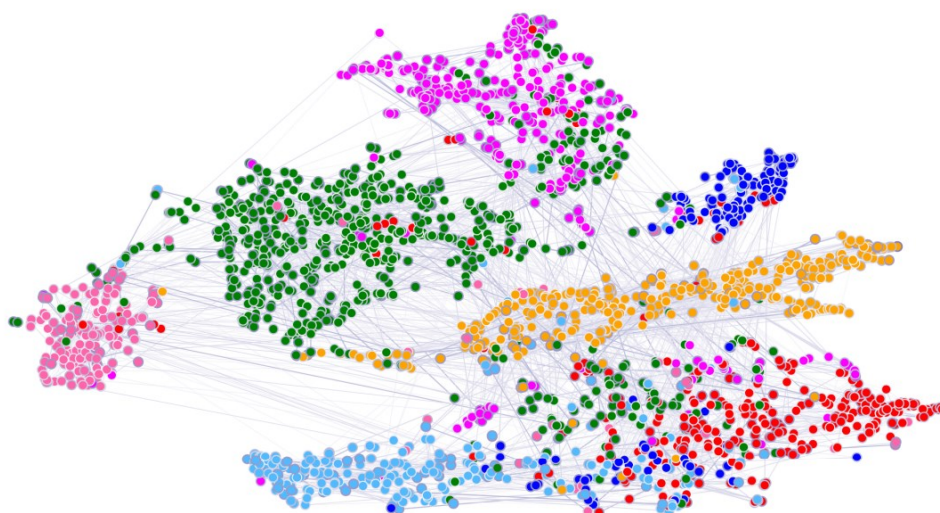
نیم سال اول ۱۴۰۱-۱۴۰۰

دیتاست Cora

مجموعه داده Cora شامل ۲۷۰۸ نشریه علمی است که در ۷ کلاس مختلف طبقه‌بندی شده‌اند. این شبکه از اسنادات شامل ۵۴۲۹ پیوند (لینک) بین نودهایش می‌باشد. هر نشریه در این مجموعه داده با یک بردار ۱۴۳۳ بعدی از 0 و 1 ها نمایش داده می‌شود که هر بعد این بردار وجود یا عدم وجود یک کلمه از دیکشنری در آن نشریه را نشان می‌دهد. ویژگی‌های این دیتاست در جدول زیر خلاصه شده است:

۲۷۰۸	تعداد نودها
۱۰۵۵۶	تعداد یال‌ها
۳/۹۰	میانگین درجه نودها
۱۴۰	تعداد نودهای آموزشی
۰/۰۵	Label rate نودهای آموزشی

شکل ۱ یک بازنمایی از گراف این دیتاست است که در مقاله‌ی اصلی GAT ارائه شده است.



شکل ۱ دیتاست Cora

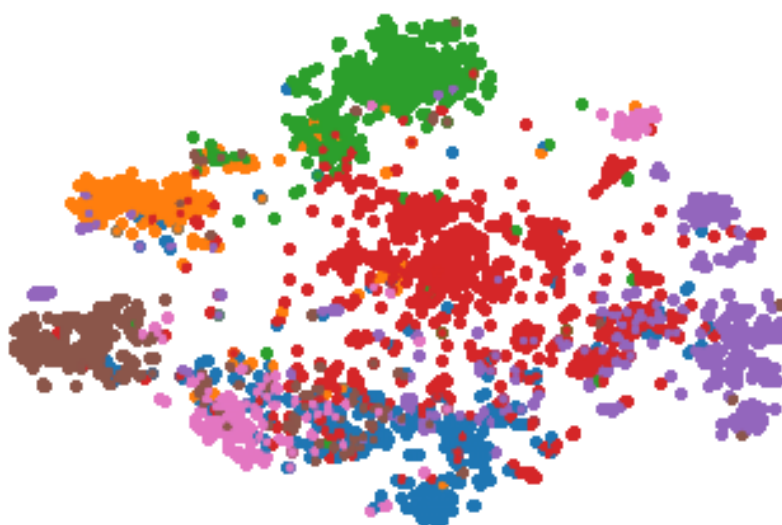
دیتاست Citeseer

مجموعه داده CiteSeer شامل ۳۳۱۲ نشریه علمی است که در ۶ کلاس طبقه‌بندی شده‌اند. شبکه اسناد شامل ۴۷۳۲ پیوند است. هر انتشار در مجموعه داده با یک بردار کلمه ۱/۰ توصیف می‌شود که نشان دهنده عدم وجود

/ وجود کلمه مربوطه در دیکشنری است. دیکشنری از ۳۷۰۳ کلمه منحصر به فرد تشکیل شده است. ویژگی‌های این دیتاست در جدول زیر خلاصه شده است:

۳۳۲۷	تعداد نودها
۹۱۰۴	تعداد یال‌ها
۲/۷۴	میانگین درجه نودها
۱۲۰	تعداد نودهای آموزشی
۰/۰۴	Label rate نودهای آموزشی

شکل ۲ یک گراف دیتاست Citeseer را نشان می‌دهد.



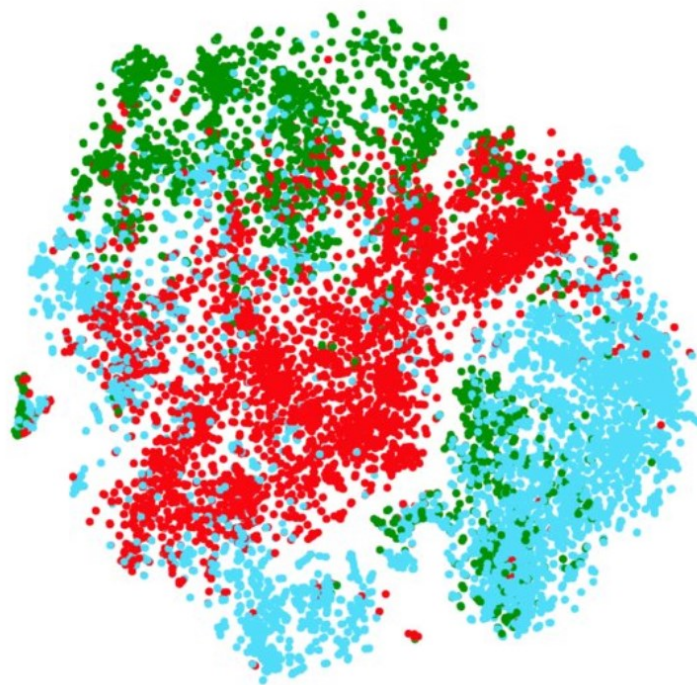
شکل ۲ دیتاست Citeseer

دیتاست Pubmed

مجموعه داده Pubmed شامل ۱۹۷۱۷ نشریه علمی از پایگاه داده PubMed مربوط به دیابت است که در یکی از سه کلاس طبقه بندی شده اند. شبکه استناد شامل ۴۴۳۳۸ پیوند است. هر انتشار در مجموعه داده توسط یک بردار کلمه وزن دار TF/IDF از یک فرهنگ لغت که از ۵۰۰ کلمه منحصر به فرد تشکیل شده است، توصیف می‌شود. ویژگی‌های این دیتاست در جدول زیر خلاصه شده است:

۱۹۷۱۷	تعداد نودها
۸۸۶۴۸	تعداد یال‌ها
۴,۵۰	میانگین درجه نودها
۶۰	تعداد نودهای آموزشی
۰/۰۰	Label rate نودهای آموزشی

شکل ۳ گراف دیتاست pubmed را نشان می‌دهد:



شکل ۳ دیتاست Pubmed

قسمت ب)

برای انجام این بخش از پروژه نیاز به تنظیم هایپرپارامترهای مدل GAT بر روی سه دیتاست مذکور است. برای تنظیم این هایپرپارامترها از مطالب گفته شده در مقاله اصلی GAT ایده گرفته شده است.

این ابرپارامترها در جدول زیر خلاصه شده‌اند:

	#output heads	Regularization factor	dropout
Cora	1	5e-4	0.6
Citeseer	1	5e-4	0.6
Pubmed	8	1e-3	0.6

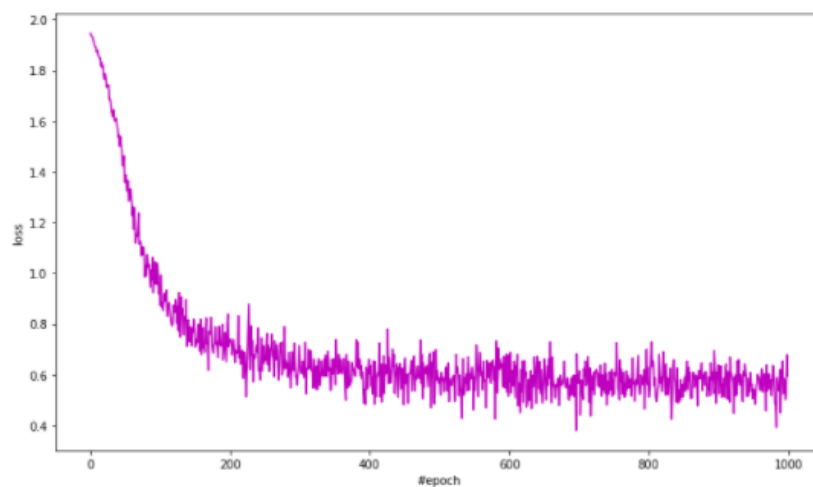
همانطور که مشاهده می‌شود ابرپارامترها برای دیتاست Cora و Citeseer دقیقا مشابه یکدیگر است. همچنین برای تجمیع سرهای مختلف دو تابع Average و concat در این کتابخانه در نظر گرفته شده است که با تنظیم ابرپارامتر Concat به مقدارهای True و False می‌توان عملکرد الحاق یا میانگین‌گیری را اعمال نمود.

در جدول زیر دقت و نوع تجمیع سرهای خروجی بر روی سه دیتاست ارائه شده است. همانطور که مشاهده می‌شود تجمیع از نوع average نتایج بهتری را حاصل کرده است.

dataset	accuracy	Aggregation type
Cora	0.8210	average
	0.8080	concatenation
Citeseer	0.7110	average
	0.7030	concatenation
Pubmed	0.7910	average
	0.7700	concatenation

برای یافتن تعداد epochهای مناسب برای آموزش مدل، نمودار loss بر حسب iteration را رسم می‌کنیم. شکل ۴ نمودار را نشان می‌دهد. از هر کجا که تغییرات Loss محسوس نبود همان تعداد epoch را برای آموزش استفاده

می‌کنیم. این نمودار برای ۱۰۰۰ تا epoch رسم شد و از حدود ۲۰۰ به بعد تغییرات چندانی در Loss مشاهده نمی‌شود بنابراین تعداد ۲۰۰ تا epoch را برای آموزش انتخاب می‌کنیم.



شکل ۴ نمودار loss بر حسب تعداد epochها

قسمت ج)

نمودارهای دقت روی داده‌های آموزشی و تست بر اساس تعداد سرها برای سه دیتاست در جدول زیر آورده شده است.

Dataset	Accuracy using GATConv																																																															
Cora	<div><p>Cora Dataset</p><table><thead><tr><th>#Heads</th><th>train accuracy</th><th>test accuracy</th></tr></thead><tbody><tr><td>1</td><td>0.970</td><td>0.775</td></tr><tr><td>2</td><td>0.985</td><td>0.785</td></tr><tr><td>3</td><td>0.980</td><td>0.815</td></tr><tr><td>4</td><td>0.980</td><td>0.810</td></tr><tr><td>5</td><td>0.980</td><td>0.815</td></tr><tr><td>6</td><td>0.985</td><td>0.810</td></tr><tr><td>7</td><td>0.980</td><td>0.820</td></tr><tr><td>8</td><td>0.985</td><td>0.815</td></tr><tr><td>9</td><td>0.980</td><td>0.825</td></tr><tr><td>10</td><td>0.980</td><td>0.815</td></tr><tr><td>11</td><td>0.985</td><td>0.825</td></tr><tr><td>12</td><td>0.985</td><td>0.820</td></tr><tr><td>13</td><td>0.985</td><td>0.815</td></tr><tr><td>14</td><td>0.985</td><td>0.810</td></tr><tr><td>15</td><td>0.985</td><td>0.820</td></tr><tr><td>16</td><td>0.990</td><td>0.825</td></tr><tr><td>17</td><td>0.990</td><td>0.815</td></tr><tr><td>18</td><td>0.985</td><td>0.825</td></tr><tr><td>19</td><td>0.985</td><td>0.825</td></tr><tr><td>20</td><td>0.990</td><td>0.820</td></tr></tbody></table></div>	#Heads	train accuracy	test accuracy	1	0.970	0.775	2	0.985	0.785	3	0.980	0.815	4	0.980	0.810	5	0.980	0.815	6	0.985	0.810	7	0.980	0.820	8	0.985	0.815	9	0.980	0.825	10	0.980	0.815	11	0.985	0.825	12	0.985	0.820	13	0.985	0.815	14	0.985	0.810	15	0.985	0.820	16	0.990	0.825	17	0.990	0.815	18	0.985	0.825	19	0.985	0.825	20	0.990	0.820
#Heads	train accuracy	test accuracy																																																														
1	0.970	0.775																																																														
2	0.985	0.785																																																														
3	0.980	0.815																																																														
4	0.980	0.810																																																														
5	0.980	0.815																																																														
6	0.985	0.810																																																														
7	0.980	0.820																																																														
8	0.985	0.815																																																														
9	0.980	0.825																																																														
10	0.980	0.815																																																														
11	0.985	0.825																																																														
12	0.985	0.820																																																														
13	0.985	0.815																																																														
14	0.985	0.810																																																														
15	0.985	0.820																																																														
16	0.990	0.825																																																														
17	0.990	0.815																																																														
18	0.985	0.825																																																														
19	0.985	0.825																																																														
20	0.990	0.820																																																														
Citeseer	<div><p>Citeseer Dataset</p><table><thead><tr><th>#Heads</th><th>train accuracy</th><th>test accuracy</th></tr></thead><tbody><tr><td>1</td><td>0.950</td><td>0.665</td></tr><tr><td>2</td><td>0.940</td><td>0.700</td></tr><tr><td>3</td><td>0.955</td><td>0.705</td></tr><tr><td>4</td><td>0.965</td><td>0.715</td></tr><tr><td>5</td><td>0.960</td><td>0.715</td></tr><tr><td>6</td><td>0.965</td><td>0.705</td></tr><tr><td>7</td><td>0.965</td><td>0.700</td></tr><tr><td>8</td><td>0.960</td><td>0.705</td></tr><tr><td>9</td><td>0.965</td><td>0.705</td></tr><tr><td>10</td><td>0.960</td><td>0.710</td></tr><tr><td>11</td><td>0.965</td><td>0.715</td></tr><tr><td>12</td><td>0.965</td><td>0.685</td></tr><tr><td>13</td><td>0.965</td><td>0.705</td></tr><tr><td>14</td><td>0.960</td><td>0.700</td></tr><tr><td>15</td><td>0.970</td><td>0.695</td></tr><tr><td>16</td><td>0.965</td><td>0.700</td></tr><tr><td>17</td><td>0.965</td><td>0.670</td></tr><tr><td>18</td><td>0.965</td><td>0.705</td></tr><tr><td>19</td><td>0.970</td><td>0.700</td></tr><tr><td>20</td><td>0.970</td><td>0.685</td></tr></tbody></table></div>	#Heads	train accuracy	test accuracy	1	0.950	0.665	2	0.940	0.700	3	0.955	0.705	4	0.965	0.715	5	0.960	0.715	6	0.965	0.705	7	0.965	0.700	8	0.960	0.705	9	0.965	0.705	10	0.960	0.710	11	0.965	0.715	12	0.965	0.685	13	0.965	0.705	14	0.960	0.700	15	0.970	0.695	16	0.965	0.700	17	0.965	0.670	18	0.965	0.705	19	0.970	0.700	20	0.970	0.685
#Heads	train accuracy	test accuracy																																																														
1	0.950	0.665																																																														
2	0.940	0.700																																																														
3	0.955	0.705																																																														
4	0.965	0.715																																																														
5	0.960	0.715																																																														
6	0.965	0.705																																																														
7	0.965	0.700																																																														
8	0.960	0.705																																																														
9	0.965	0.705																																																														
10	0.960	0.710																																																														
11	0.965	0.715																																																														
12	0.965	0.685																																																														
13	0.965	0.705																																																														
14	0.960	0.700																																																														
15	0.970	0.695																																																														
16	0.965	0.700																																																														
17	0.965	0.670																																																														
18	0.965	0.705																																																														
19	0.970	0.700																																																														
20	0.970	0.685																																																														
Pubmed	<div><p>Pubmed Dataset</p><table><thead><tr><th>#Heads</th><th>train accuracy</th><th>test accuracy</th></tr></thead><tbody><tr><td>1</td><td>0.900</td><td>0.720</td></tr><tr><td>2</td><td>0.910</td><td>0.700</td></tr><tr><td>3</td><td>0.920</td><td>0.720</td></tr><tr><td>4</td><td>0.935</td><td>0.745</td></tr><tr><td>5</td><td>0.935</td><td>0.755</td></tr><tr><td>6</td><td>0.935</td><td>0.745</td></tr><tr><td>7</td><td>0.935</td><td>0.755</td></tr><tr><td>8</td><td>0.935</td><td>0.775</td></tr><tr><td>9</td><td>0.945</td><td>0.775</td></tr><tr><td>10</td><td>0.945</td><td>0.750</td></tr><tr><td>11</td><td>0.955</td><td>0.770</td></tr><tr><td>12</td><td>0.945</td><td>0.765</td></tr><tr><td>13</td><td>0.945</td><td>0.775</td></tr><tr><td>14</td><td>0.955</td><td>0.775</td></tr><tr><td>15</td><td>0.945</td><td>0.770</td></tr><tr><td>16</td><td>0.955</td><td>0.780</td></tr><tr><td>17</td><td>0.955</td><td>0.770</td></tr><tr><td>18</td><td>0.945</td><td>0.765</td></tr><tr><td>19</td><td>0.955</td><td>0.780</td></tr><tr><td>20</td><td>0.955</td><td>0.775</td></tr></tbody></table></div>	#Heads	train accuracy	test accuracy	1	0.900	0.720	2	0.910	0.700	3	0.920	0.720	4	0.935	0.745	5	0.935	0.755	6	0.935	0.745	7	0.935	0.755	8	0.935	0.775	9	0.945	0.775	10	0.945	0.750	11	0.955	0.770	12	0.945	0.765	13	0.945	0.775	14	0.955	0.775	15	0.945	0.770	16	0.955	0.780	17	0.955	0.770	18	0.945	0.765	19	0.955	0.780	20	0.955	0.775
#Heads	train accuracy	test accuracy																																																														
1	0.900	0.720																																																														
2	0.910	0.700																																																														
3	0.920	0.720																																																														
4	0.935	0.745																																																														
5	0.935	0.755																																																														
6	0.935	0.745																																																														
7	0.935	0.755																																																														
8	0.935	0.775																																																														
9	0.945	0.775																																																														
10	0.945	0.750																																																														
11	0.955	0.770																																																														
12	0.945	0.765																																																														
13	0.945	0.775																																																														
14	0.955	0.775																																																														
15	0.945	0.770																																																														
16	0.955	0.780																																																														
17	0.955	0.770																																																														
18	0.945	0.765																																																														
19	0.955	0.780																																																														
20	0.955	0.775																																																														

همانطور که مشاهده می‌شود برای هر سه دیتاست مقدار ماکزیمم دقت در حدود بین ۶ تا ۱۰ برای تعداد سرها اتفاق می‌افتد. اگر تعداد سرها از این عدد بیشتر باشد مدل پیچیده می‌شود و اطلاعات زیادی را باید تجمیع کند و اگر کمتر از این تعداد باشد مدل ساده است و به اندازه کافی قوی نیست.

به همین دلیل در مقاله GAT نیز تعداد بهینه برای تعداد سرها برای هر سه دیتاست عدد ۸ در نظر گرفته شده است. ما نیز در ادامه برای انجام بخش‌های این پروژه عدد ۸ را برای تعداد سرها در نظر می‌گیریم.

قسمت د)

برای انجام این بخش در کد منبع GATconv بخشی که مربوط به محاسبه الحاق می‌باشد را Generalize می‌کنیم تا بتوان هر تابعی را به جای آن قرار داد. قسمتی از کد GATConv که مربوط به این بخش است در شکل زیر نشان داده شده است:

```
# Next, we compute node-level attention coefficients, both for source
# and target nodes (if present):
alpha_src = (x_src * self.att_src).sum(dim=-1)
alpha_dst = None if x_dst is None else (x_dst * self.att_dst).sum(-1)
alpha = (alpha_src, alpha_dst)
```

در واقع باید خط سوم این تکه کد را به شکل زیر تغییر بدهیم:

```
# Next, we compute node-level attention coefficients, both for source
# and target nodes (if present):
alpha_src = (x_src * self.att_src).sum(dim=-1)
alpha_dst = None if x_dst is None else (x_dst * self.att_dst).sum(-1)
alpha = self.agg_func(alpha_src, alpha_dst)
```

که در آن Self.agg_func یک تابع است که به عنوان ورودی به مدل داده می‌شود و ما برای انجام این بخش توابع min, max, sum, mean, hadamard را تعریف کرده و به عنوان ورودی می‌دهیم.

دقت به دست آمده بر روی مجموعه داده‌های مختلف و با استفاده از توابع مختلف در جدول زیر آورده شده است:

Pubmed	Citeseer	Cora	
0.7810	0.6990	0.8180	min
0.7920	0.7130	0.8150	max
0.7790	0.7020	0.8320	sum
0.7910	0.7050	0.8190	mean
0.7870	0.7130	0.8200	hadamard

برای انجام concat از آنجایی که عمل concat ابعاد را ۲ برابر می‌کند یک راه حل این است که یک تبدیل خطی روی alpha_src و alpha_dst قبل از concatenation بزنیم تا ابعاد هر کدام نصف شود:

```
self.lin_batch_reduce = Linear(batch_dim, int(batch_dim/2), bias=False, weight_initializer='glorot')
```

سپس تغییرات روی مقادیر آلفا را به این صورت اعمال می‌کنیم:

```

alpha_src = (x_src * self.att_src).sum(dim=-1).transpose(0, -1)
alpha_src = self.lin_batch_reduce(alpha_src).transpose(-1, 0)
alpha_dst = None if x_dst is None else (x_dst * self.att_dst).sum(-1).transpose(0, -1)
if alpha_dst is not None: alpha_dst = self.lin_batch_reduce(alpha_dst).transpose(-1, 0)
alpha = self.agg_func(alpha_src, alpha_dst)

```

قسمت ه)

در این بخش هر یک از سرهای توجه از یک تابع مجزا استفاده می‌کند. به همین دلیل روی هر کدام از هدها یکی از توابع پنج‌گانه گفته شده را اعمال کرده و در نهایت نتایج را کنار هم قرار می‌دهیم. برای انجام این بخش یک فانکشن به صورت زیر تعریف شده است:

```

def five_head_func(alpha_src, alpha_dst):
    alpha_1 = mymin(alpha_src[:,0:1], alpha_dst[:,0:1])
    alpha_2 = mymax(alpha_src[:,1:2], alpha_dst[:,1:2])
    alpha_3 = mean(alpha_src[:,2:3], alpha_dst[:,2:3])
    alpha_4 = hadamard(alpha_src[:,3:4], alpha_dst[:,3:4])
    alpha_5 = mysum(alpha_src[:,4:5], alpha_dst[:,4:5])
    tensors= [alpha_1,alpha_2,alpha_3,alpha_4,alpha_5]
    alpha = torch.cat(tensors, 1)

    return alpha

```

و این تابع به جای `self.agg_func` که در قسمت قبل توضیح داده شد قرار می‌گیرد. نتایج حاصل به صورت جدول زیر است:

Dataset	Accuracy
Cora	0.8360
Citeseer	0.7120
Pubmed	0.7870

همانطور که مشاهده می‌شود دقت برای دو دیتاست اول مقدار کمی افزایش و برای دیتاست سوم کمی کاهش پیدا کرده است.

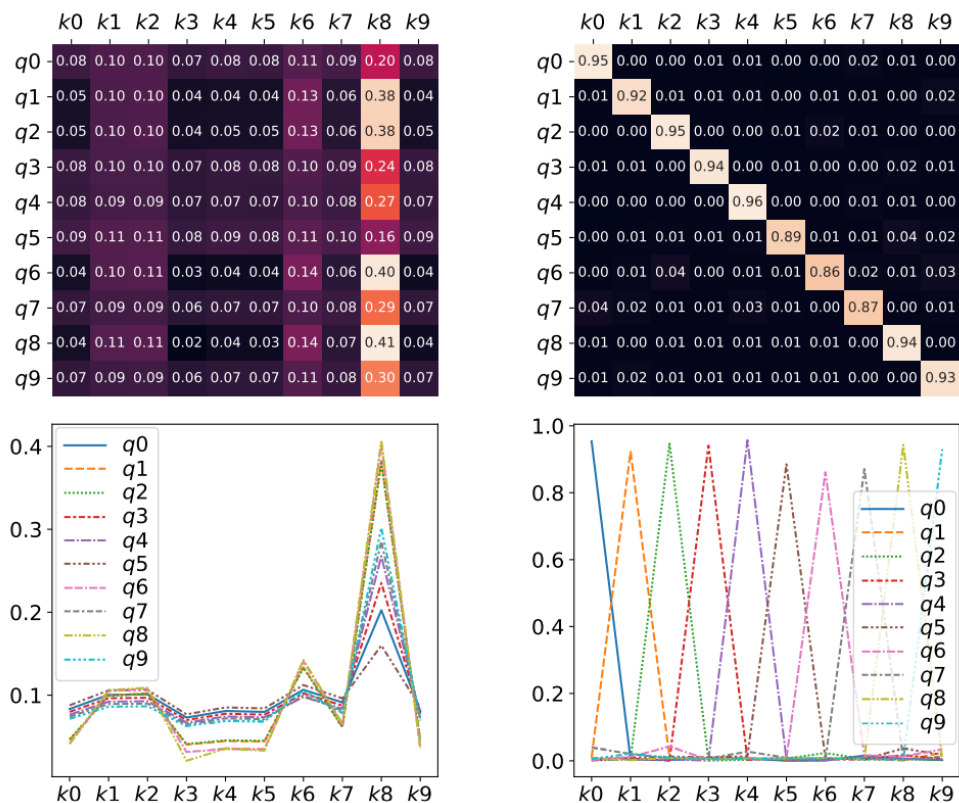
در GAT، هر گره با داشتن بازنمایی خودش به عنوان کوئری، به همسایگان خود توجه می کند. در این مقاله نشان داده شده است که GAT نوع بسیار محدودی از توجه را محاسبه می کند و در واقع رتبه بندی امتیازات توجه به نود کوئری (مورد سوال) بستگی ندارد. این نوع توجه محدود به "توجه ایستا" معروف است و مقاله آن را از "توجه پویا" که قدرت بیان آن بسیار بالاتر از توجه ایستا است، متمایز می کند.

در توجه ایستا برای هر نودی که کوئری باشد تابع توجه نسبت به امتیازات همسایه ها یکنوا است. این بدان معناست که رتبه بندی ضرایب توجه در بین تمامی نودهای گراف به اشتراک گذاشته می شود و به نود کوئری بستگی ندارد.

این مسئله در شکل ۵ که از مقاله مذکور اقتباس شده است، نشان داده شده است. با توجه به این شکل، در یک نمودار دو بخشی کامل از گره های کوئری که با آن ها را با $\{q_0, \dots, q_9\}$ نشان می دهیم و "گره های کلید" که آن ها را با $\{k_0, \dots, k_9\}$ نشان می دهیم، قسمت سمت چپ شکل GAT استاندارد را نمایش می دهد که رتبه بندی ضرایب توجه برای همه گره های گراف به اشتراک گذاشته می شود و به گره کوئری بستگی ندارد. به عنوان مثال، تمام کوئری ها در GAT استاندارد بیشتر به کلید ۸ (k_8) مربوط می شوند. در مقابل، GATv2 که در قسمت سمت راست شکل است، در واقع می تواند توجه پویا را محاسبه کند، به طوری که هر کوئری دارای رتبه بندی متفاوتی از ضرایب توجه کلیدها است.

¹ Static attention

² monotonic



شکل ۵ مقایسه توجه ایستا و پویا

به دلیل اینکه GAT ها از مکانیزم توجه ایستا استفاده می کنند، مسائل گرافی ساده‌ای وجود دارد که نمی توانند آن‌ها را بیان کنند. نشان داده شده است که توجه ایستا حتی از فیت شدن داده‌های آموزشی در GAT جلوگیری می‌کند. طبق تعریف توجه ایستا که در مقاله ارائه شده است، توجه ایستا بسیار محدود است زیرا تابع $f \in F$ یک کلید دارد که همیشه بدون توجه به کوئری انتخاب می‌شود. چنین تابعی نمی تواند موقعیت‌هایی را مدل کنند که در آن کلیدهای مختلف ارتباط متفاوتی با کوئری‌های مختلف دارند.

برای حذف این محدودیت، این مقاله یک راه حل ساده را با اصلاح ترتیب عملیات معرفی می‌کند و GATv2 را پیشنهاد می‌دهد که به شدت گویاتر از GAT است. محاسبه‌ی امتیاز توجه در GAT به صورت زیر انجام می‌شود:

$$e(h_i, h_j) = \text{LeakyReLU}(a^T \cdot [Wh_i || Wh_j])$$

در واقع مشکل اصلی در GAT از این جا نشأت می‌گیرد که لایه‌های W و a پشت سر هم اعمال می‌شوند و چون هر دو تبدیل خطی هستند، طبق خواصی که تبدیل خطی دارد می‌توانند به یک لایه منفرد ادغام^۳ شوند. برای حل این مشکل GATv2 یک راه حل ساده ارائه می‌دهد و آن هم این است که لایه‌ی a بعد از غیرخطی ساز

^۳ collapse

(LeakyReLU) اعمال شود و لایه‌ی W بعد از Concatenation اعمال شود. بنابراین فرمولی که GATv2 از آن برای محاسبه توجه استفاده می‌کند به این صورت است:

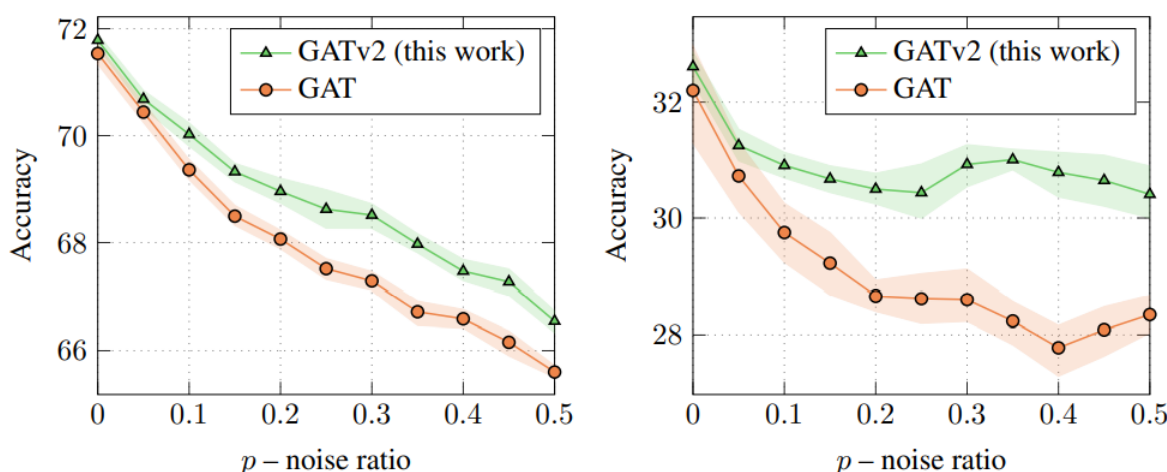
$$e(h_i, h_j) = a^T \text{LeakyReLU}(W \cdot [h_i || h_j])$$

این اصلاح ساده تفاوت قابل توجهی در قدرت بیان تابع توجه ایجاد می‌کند. پیچیدگی محاسباتی GATv2 برابر با GAT می‌باشد: $O(|V|dd' + |\mathcal{E}|d')$ که البته چون در GAT لایه‌های خطی می‌توانند ادغام شوند، محاسبات در GAT سریع‌تر صورت می‌گیرد.

یک تفاوت دیگر GATv2 نسبت به GAT که در واقع یک مزیت است، این است که GATv2 نسبت به نویز یال‌ها مقاوم‌تر است زیرا مکانیسم توجه پویا به آن کمک می‌کند تا یال‌های نویزی را کاهش دهد در حالی که عملکرد GAT استاندارد در صورت وجود چنین یال‌هایی بسیار کاهش پیدا می‌کند.

این مسئله در شکل زیر نشان داده شده است. نویسندگان مقاله بر روی دو دیتاست مقاومت توجه پویا و ایستا در برابر نویز را آزمودند و نتایج آن در دو نمودار گزارش شده است. دقت پیش‌بینی گره بر حسب نسبت نویز در این نمودارها گزارش شده است. همانطور که مشاهده می‌شود با افزایش میزان نویز دقت در هر دوی GAT و GATv2 کاهش پیدا می‌کند اما میزان و شیب این کاهش در GATv2 بسیار کمتر است.

نویسندگان ادعا می‌کنند که توانایی انجام توجه پویا به مدل‌ها کمک می‌کند تا بین یال‌های داده شده و یال‌های نویز تمایز قائل شوند. در مقابل، GAT نمی‌تواند بین یال‌ها تمایز قائل شود، زیرا گره‌های مبدا و مقصد را به طور جداگانه امتیاز می‌دهد. این نتایج به وضوح استحکام توجه پویا را نسبت به توجه ایستا در مواجهه با نویز، که در واقعیت هم رایج هستند، نشان می‌دهد.



قسمت ز)

دقت‌های به دست آمده برای سه دیتاست با استفاده از GAT و GATv2 در جدول زیر آورده شده است. همانطور که مشاهده می‌شود برای دو دیتاست Cora و Citeseer دقت‌ها در GATv2 نسبت به GAT مقدار کمی افزایش داشته‌اند اما در مورد دیتاست Pubmed دقت‌ها در GATv2 نسبت به GAT مقدار کمی کاهش داشته‌اند.

dataset	GATv2 Accuracy	GAT Accuracy	Aggregation
Cora	0.8220	0.8210	Avg
	0.8300	0.8080	concat
Citeseer	0.7120	0.7110	Avg
	0.7080	0.7030	concat
Pubmed	0.7880	0.7910	Avg
	0.7550	0.7700	concat

نمودارهای دقت روی داده‌های آموزشی و تست برای تعداد هدهای از ۱ تا ۲۰ با استفاده از GAT و GATv2 در جدول صفحه بعد آورده شده است. همانطور که مشاهده می‌شود به طور کلی دقت‌ها (به خصوص در دو دیتاست اول) با استفاده از روش GATv2 بالاتر از GAT هستند.

