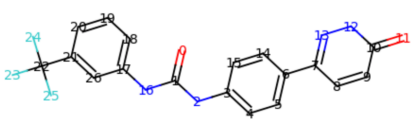
۱- در مدل ارائه شده برای پیش‌بینی مقدار K_d برخلاف Baselineها و روش‌های قبلی ارائه شده از یادگیری عمیق استفاده شده است. در این روش ارائه شده پروتین‌ها را که به صورت رشته‌ای از حروف هستند، هر حرف آن را معادل یک عدد قرار داده و ورودی می‌دهند. داروها با فرمت SMILES می‌باشند که این را نیز می‌توان به صورت رشته‌ای از کارکترها در نظر گرفته و هر کاراکتر را معادل یک عدد بگیریم و ورودی بدهیم. هر کدام از ورودی‌های گفته شده را ابتدا به یک شبکه CNN ورودی می‌دهند تا featureهای آن‌ها را استخراج کند. سپس featureهای هر دسته دارو و پروتین که می‌خواهیم میزان سازگاری آن‌ها را تشخیص دهیم، باهم ترکیب می‌کنیم و از ۳ لایه‌ی Fully Connected عبور می‌دهیم. در نهایت خروجی انتظار داریم عدد مورد نظر باشد و Loss آن را MSELoss تعریف می‌کنیم.
از جمله نقاط قوت آن می‌توان گفت که به جای استفاده از یک سری feature مشخص استخراج شده از داده برای پیش‌بینی، از داده‌ی خام استفاده کرده‌است. در واقع featureهای مفید را با استفاده از CNN یاد گرفته است و این باعث پیش‌بینی بهتر مخصوصا در داده‌های حجیم‌تر شده است. همانطور که در نتیجه‌ها مشخص است استفاده از CNN برای یادگیری فضای featureها موجب ارتقا بیشتر در دیتاست Kiba که ۴ برابر دیتاست Davis می‌باشد، چون قابلیت پوشش کامل‌تر ویژگی‌های متفاوت و یادگیری آن‌ها را دارد. همانطور که در مقاله هم به عنوان future work مطرح شده است، یکی از مشکلات این شبکه توجه نکردن به خاصیت سری بودن پروتین‌ها و استفاده از CNN ساده به جای LSTM که برای داده‌های ترتیبی مناسب‌تر است، می‌باشد. از مشکلات دیگر این شبکه می‌توان به توجه نکردن به ساختار مولکولی داروها و ارتباط بین اتم‌ها هر مولکول می‌باشد که می‌توان اطلاعات زیادی به شبکه بدهد.

۲و۳- اولویت من در ارائه‌ی این مدل بر طرف کردن مشکل دوم مطرح شده می‌باشد. در مقاله‌ی GraphDTA، ۴ روش Graph Based برای حل این مشکل ارائه شده است. ایده‌ای کلی بدین صورت است که کلیت مدل مانند مدل DeepDTA می‌باشد، اما به جای آن که دارو را به صورت یک رشته ورودی دهیم، ساختار مولکولی آن را حفظ کرده و آن را به صورت گرافی از اتم‌ها ورودی دهیم. گراف مطرح شده، بدین صورت طراحی می‌شود که هر اتم به صورت یک رشته‌ی ۷۸تایی encode می‌شود.
اما علاوه بر رشته‌ی encode شده یک ماتریس ۲ بعدی با ابعاد 2*N وجود دارد که اتصال بین اتم‌ها را نشان می‌دهد. به طور مثال ساختار مولکولی یک دارو در شکل روبرو نشان داده می‌شود.



یکی از روش‌ها GCNNet می‌باشد که به جای استفاده از لایه‌های CNN از ۳ لایه GCN و یک لایه‌ی Global Max Pooling استفاده می‌شود. روش دوم GATNet می‌باشد که بر اساس self-attention یادگیری را انجام می‌دهد. در این روش از ۲ لایه‌ی GAT و یک لایه‌ی Global Max Pooling استفاده می‌شود. روش سوم GINNet می‌باشد که گفته می‌شود این روش به بیشترین حد discriminative power بین GNNها می‌رسد.در این روش از ۵ لایه‌ی GINConv استفاده می‌شود که بعد هر کدام یک لایه‌ی batch normalization می‌آید استفاده شده است و در نهایت از یک لایه‌ی Global Max Pooling استفاده می‌شود. روش آخر که یک روش ترکیبی از GCN و GAT می‌باشد و GAT-GCN Net نامیده می‌شود. در این روش به ترتیب از یک لایه‌ی GAT، یک لایه‌ی GAN و یک لایه‌ی Global Max Pooling استفاده می‌شود.

۴- کد مدل‌های گفته شده و ۸ تا از مدل‌های pretraind شده با هر روش و روی هر سری داده در فولدر pretrained قرار دارد. در کد زده شده داده‌های validation بر اساس setting اعلام شده نبود و به صورت

رندم ۱/۵ داده یادگیری را به validation اختصاص می‌دادند. طبق اعداد گفته شده در مقاله MSE و CI روی هر داده در مقایسه با روش‌های دیگر به صورت زیر می‌باشد.

Method	Protein rep.	Compound rep.	CI	MSE
Baseline models				
DeepDTA	Smith-Waterman	Pubchem-Sim	0.790	0.608
DeepDTA	Smith-Waterman	1D	*0.886*	0.420
DeepDTA	1D	Pubchem-Sim	0.835	0.419
KronRLS	Smith-Waterman	Pubchem-Sim	0.871	0.379
SimBoost	Smith-Waterman	Pubchem-Sim	0.872	0.282
DeepDTA	1D	1D	0.878	*0.261*
WideDTA	1D + PDM	1D + LMCS	*0.886*	0.262
Proposed model - **GraphDTA**				
GCN [17]	1D	Graph	0.880	**0.254**
GAT_GCN	1D	Graph	0.881	**0.245**
GAT [37]	1D	Graph	**0.892**	**0.232**
GIN [40]	1D	Graph	**0.893**	**0.229**

(a) For Davis dataset, sorted by MSE. Italics: best for baseline models, bold: better than baselines.

Method	Protein rep.	Compound rep.	CI	MSE
Baseline models				
DeepDTA	1D	Pubchem-Sim	0.718	0.571
DeepDTA	Smith-Waterman	Pubchem-Sim	0.710	0.502
KronRLS	Smith-Waterman	Pubchem-Sim	0.782	0.411
SimBoost	Smith-Waterman	Pubchem-Sim	0.836	0.222
DeepDTA	Smith-Waterman	1D	0.854	0.204
DeepDTA	1D	1D	0.863	0.194
WideDTA	1D + PDM	1D + LMCS	*0.875*	*0.179*
Proposed model - **GraphDTA**				
GAT [37]	1D	Graph	0.866	0.179
GIN [40]	1D	Graph	**0.882**	**0.147**
GCN [17]	1D	Graph	**0.889**	**0.139**
GAT_GCN	1D	Graph	**0.891**	**0.139**

(b) For Kiba dataset, sorted by MSE. Italics: best for baseline models, bold: better than baselines.

Table 1: Prediction performance. Baseline results are from [25]. For our proposed method, same settings as with the referenced methods, e.g., train/test splits, was used.

تغییرات لازم را در ایجاد دیتاست طبق گفته‌ی سوال انجام داده شد و به‌دلیل محدودیت منابع مدل GCN را ۱۰۰۰ epoch رو هر کدام از داده‌های davis و کیبا یادگیری انجام داده شد. نتایج به صورت زیر بود.

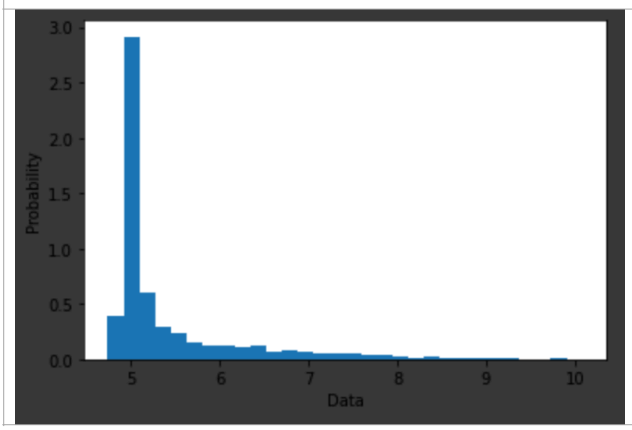
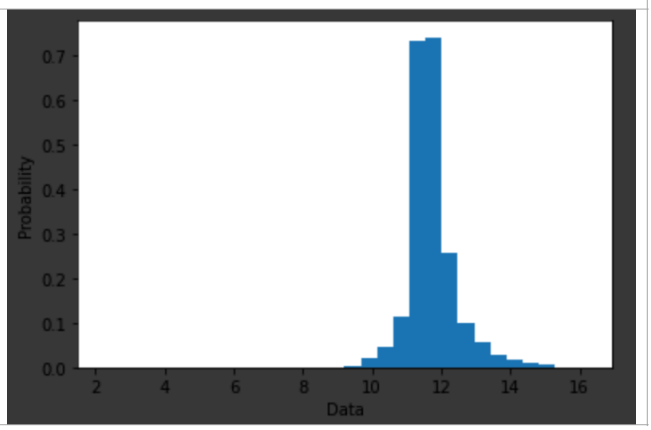
	CI	MSE
Davis	0.8672866388	0.2922719
Kiba	0.8260702233	0.24018808

همانطور که مشاهده می‌شود، به دلیل محدودیت منابع به مقدار گفته شده در مقاله نرسید، مخصوصا برای داده‌ی Kiba.

۵- با مشاهده‌ی نتایج می‌توانیم بفهمیم در حالت کلی داده به سمت تشخیص عدم سازگاری بایاس دارد و این در Davis شدیدتر است. از جمله دلیل‌های آن می‌توان به نا متقارنی داده و اینکه حجم عظیمی از داده ها ناسازگارند اشاره کرد.

	Accuuracy	Sensivity	Specifity	F1 score
Davis	0.94	0.98214674	0.53237410	0.97021185
Kiba	0.87	0.93108964	0.64411764	0.92005184

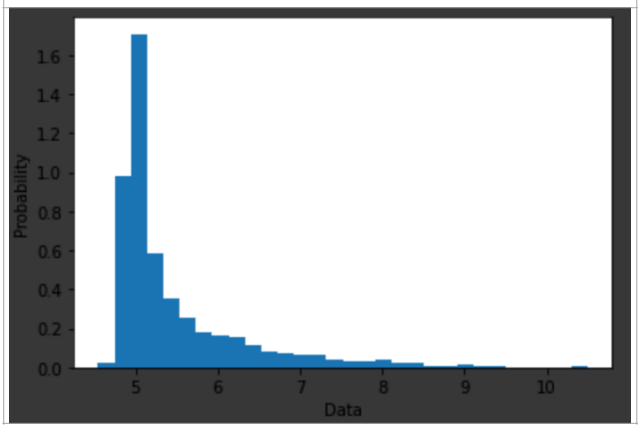
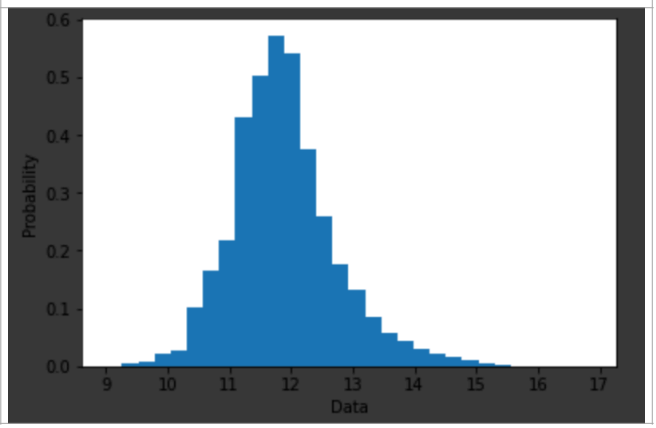
۶- مورد مطرح شده در بالا به صورت کامل در هیستگرام دیده می‌شود

Davis	Kiba
	

برای حل این مشکل یکی از راه‌های ممکن استفاده از weighted sampling می‌باشد که در این روش dataloader سعی می‌کند از داده‌هایی که تعداد کمتر دارد بیشتر سمپلگیری انجام دهد در نتیجه همه‌ی داده‌ها را به دفعات یکسان دیده و این مشکل unbalanced بودن را حل کند. تغییرات را اعمال کردیم و در نتیجه در نتیجه‌ی نهایی در هر داده داده افت زیادی را می‌بینیم، اما specifity هر دو به میزان قابل توجهی افزایش یافته که خود نشان دهنده ی تاثیر گذاری روش ما بود. همچنین چون یک trade off بین specifity و sensivity وجود دارد به تبع این اتفاق sensivity هم کاهش یافته است. در هیستگرام رسم شده هم تغییر توزیع نتایج کاملا مشهود می‌باشد.

	CI	MSE
Davis	0.8478679542	0.35140294
Kiba	0.7523496431	0.46476233

	Accuuracy	Sensivity	Specifity	F1 score
Davis	0.94	0.97713912	0.56834532	0.96922578
Kiba	0.77	0.77055473	0.77916666	0.84296363

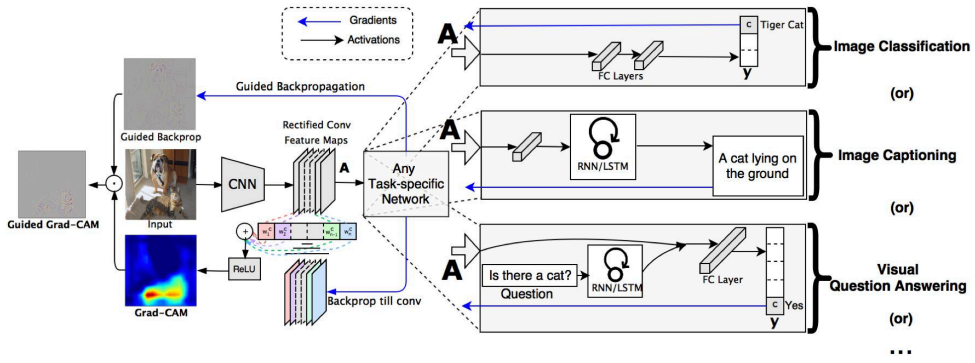
Davis	Kiba
	

۷- بهترین مدل مدل یادگرفته شده بدون weighted sampling روی داده davis می‌باشد.

۸- saliency map: در این روش گرادیان خروجی نسبت به ورودی گرفته می‌شود و در نهایت هر عددی نسبت به هر عضو داده (هر حرف پروتین‌ها و هر اتم مولکول‌ها) برابر اندازه خود ورودی‌ها نسبت داده می‌شود که بالاتر بودن این عدد نشان دهنده‌ی توجه بیشتر شبکه ی یادگرفته شده به آن عضو می‌باشد.

Guided Back Propagation: منطق طراحی این روش این است که در هنگام گرادیان گرفتن، تاثیرهای مثبت را در نظر بگیریم و تاثیرهای منفی را می‌توان برابر صفر قرار داد. به همین منظور به جای لایه‌های relu در گرادیان گیری شبکه نسبت به ورودی از لایه‌ی relu برعکس استفاده می‌کند. همچنین از این روش می‌توان به عنوان روش unsupervised segmentation استفاده کرد.

Guided Grad-Cam: در Grad-Cam ساده بستگی به نوع تسک classification، segmetation یا visual questioning answering لایه‌های اخر را در نظر نمی‌گیرند و نسبت آخرین لایه convolution گرادیان را حساب می‌کنند. سپس میانگین وزن‌دار لایه‌ها را حساب کرده و از لایه‌ی relu عبور داده و ماسک را بدست می‌آورند. وزن هر لایه متناسب با گرادیان محاسبه شده است. Guided Grad-Cam ضرب این ماسک با ماسک Guided Back Propagation می باشد.



LRP: در این روش در حال back propagation به هر لایه یک عدد R متناسب فرمول $R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$، اختصاص می‌دهند. که a_j مقدار خروجی یک لایه از اکتیویشن فانکشن و w_j همان وزن‌های آن لایه است. این فرمول را برای لایه‌های متوالی محاسبه می‌کنیم تا به لایه‌ی input برسیم و اعداد بدست آمده همان ماسک مورد نظر است.

توضیحات فایل‌های ارسال شده:

کد مدل طراحی شده در gcn.py قرار دارد و یادگیری و کدهای ارزیابی به وسیله جوپیتر نوتبوک در فایل project_phase2.ipynb قرار دارد.
۴ مدل یادگرفته شده روی davis و kiba در حالت وزن‌دار و غیر وزن‌دار به همراه نتایج به ترتیب در فایل‌های model. و .csv قرار گرفتند. همچنین مدل های یادگرفته شده روی هر ۴ مدل توسط نویسنده‌های مقاله را در فایل pretrained قرار دارد که امکان لود و دیدن نتایج وجود دارد.
در فایل‌های نتایج csv. معیارهای به ترتیب rmse، mse، pearson، spearman و ci قرار گرفته شده است.
فایل اسکریپت Main.py مطابق گفته ساخته شده و در فایلresult_GCNNet_davis_test.csv نتیجه‌ی داده تست ارسال شده در piazza قرار دارد.
در فایل data داده‌های preprocess شده موجود می‌باشند.
برای انجام training مجدد با عوض کردن WEIGHTED به True یا False می‌توانید در ۲ حالت مختلف یاد بگیرید. همچنین با انتخاب یک dataset از datasets می‌توانید یادگیری روی یک دیتا انجام دهید.
چون امکان آپلود کامل وجود نداشت برای راحتی کار پیشنهاد می‌کنم از لینک google drive زیر استفاده کنید که همه‌ی فایل‌های مذکور در آن قرار دارند.
https://drive.google.com/drive/folders/1-ITb9mNqsxnMdNzy_nnaMuLxF_YpJGbP?usp=sharing