COMP396

SMILES string molecular descriptors (continued..)

Summary

1. Recall
2. Description
3. ChemVAE
4. Training visualizations
5. MORE DATA ?
6. Comparisons
7. Questions
5. Next steps?

Recall:

- During our last meeting, I presented my SMILES + CNN and LSTM findings/results and they were below par. https://docs.google.com/document/d/1k9qPo0a1o25bREa8aOYaFWFB1Wk_QIeEVd6wr7wMnzc/edit?usp=sharing
- You advised me to :

1) double-check my training results
2) use visualization dashboard (Comet.ml or Tensorboard) to attempt to debug the issue in the training.
3) Focus on the SMILES + LSTM before moving to DGI

Description of SMILES String:

SMILES (Simplified molecular-input line-entry system) strings are a compact way of representing molecules. In this project, we use SMILES strings in order to predict target properties from the QM9 benchmark and the Zinc dataset. The SMILES that are retrieved from the XYZ files are canonicalized and thus have a unique representation.

The main advantage of using SMILES string-based molecular descriptors is that they are less sophisticated then graph neural networks and perform relatively well.

SMILES representations can also be modified to include chiral indications. However, we only consider non-isomeric molecules from the QM9 dataset in this milestone.

We experiment with two different ways of building a molecular descriptor with the SMILES strings.

With Pytorch, we used an embedding layer that maps that integer indices to dense vectors.

ChemVae

Smiles String-based molecular descriptor Inspiration from -> Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules https://arxiv.org/pdf/1610.02415.pdf

It is a method to convert discrete representations of molecules to and from a multidimensional continuous representation for efficient molecule generation. The VAE autoencoder may also be jointly trained with property prediction to help shape the latent space.

The search space of molecular data is usually large, discrete, and unstructured.

Important points:

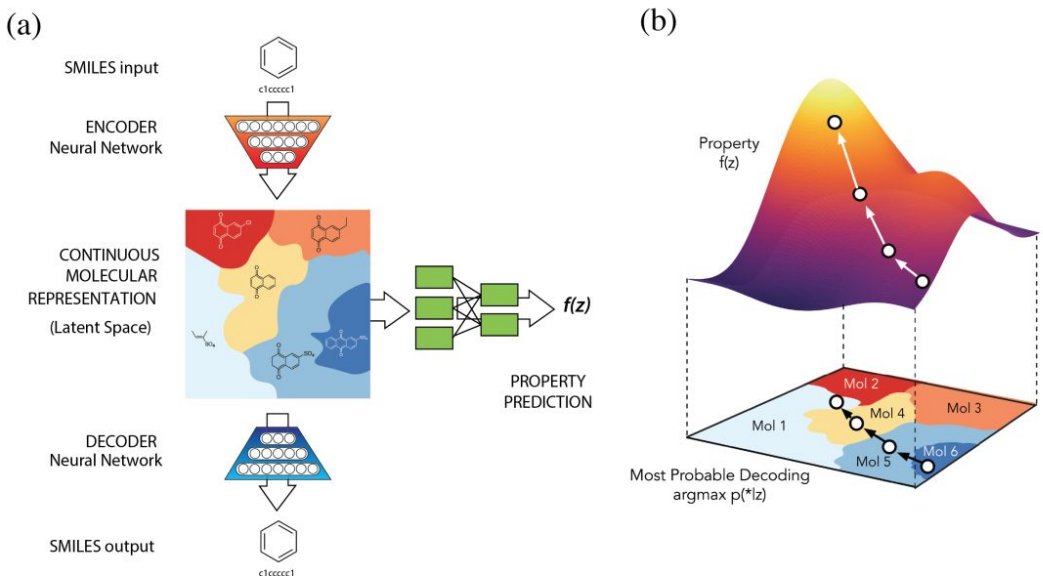
1st experiment : Constrained optimization-autoencoder + MLP jointly on a property prediction task with smiles

- Authors also tested Inchi(International Chemical Identifier), however, the generalization ability was worse due to the complexity of the syntax.

2nd experiment: Unconstrained optimization-VAE autoencoder used (for latent space to correspond to valid decoding) + RNN(GRU) & CNN.

- Comparison of the validity of generated molecules with a genetic algorithm

3rd experiment: property prediction task - see table on comparison section.



Benchmarks:

QM9: Already covered.

ZINC: The zinc 250k dataset is retrieved from the zinc database which is a free public resource for ligand discovery. The database contains over twenty million commercially available molecules in biologically relevant representations that may be downloaded in popular ready-to-dock formats and subsets https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4658288/.

The smiles string are mapped to the following target properties

1) Water-octanol partition coefficient (logP)(also known as lipophilicity)
2) Synthetic accessibility score (SAS)
3) Qualitative Estimate of Drug-likeness (QED)

Training visualization:

Dataset: QM9

Loss: Mean squared error (MSE)

Error: Mean absolute error (MAE)

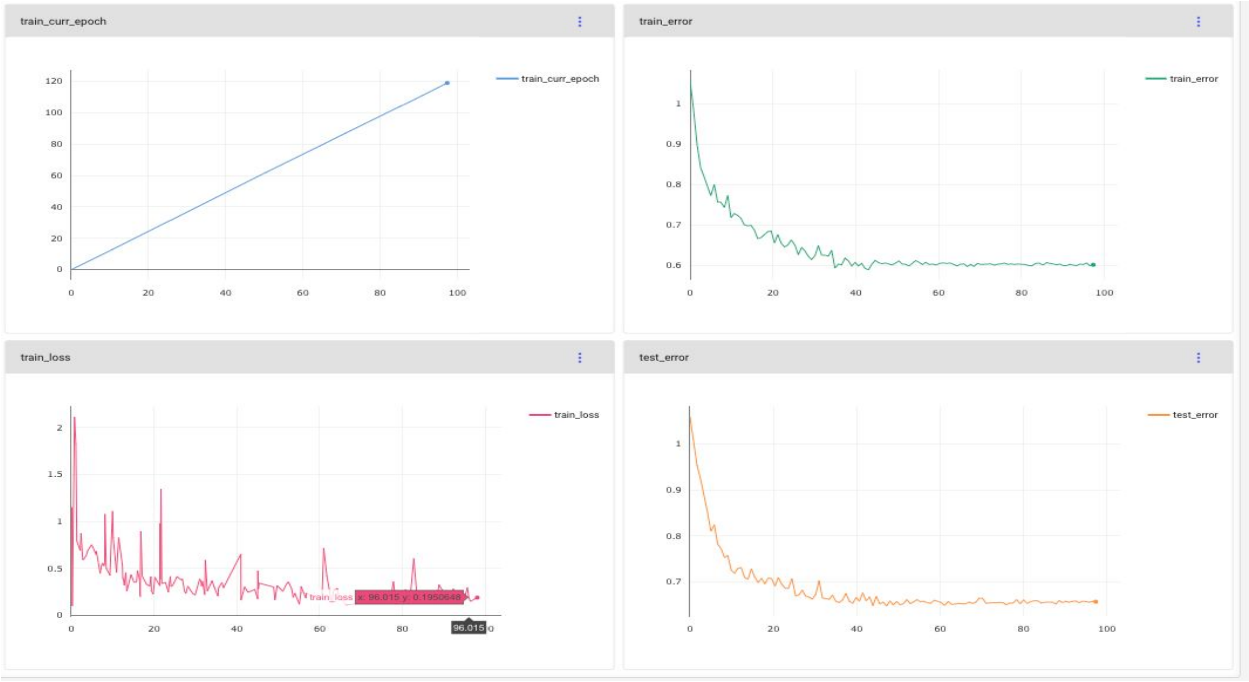
Stochastic optimization: Adam + lr scheduler on plateau by a factor of 2

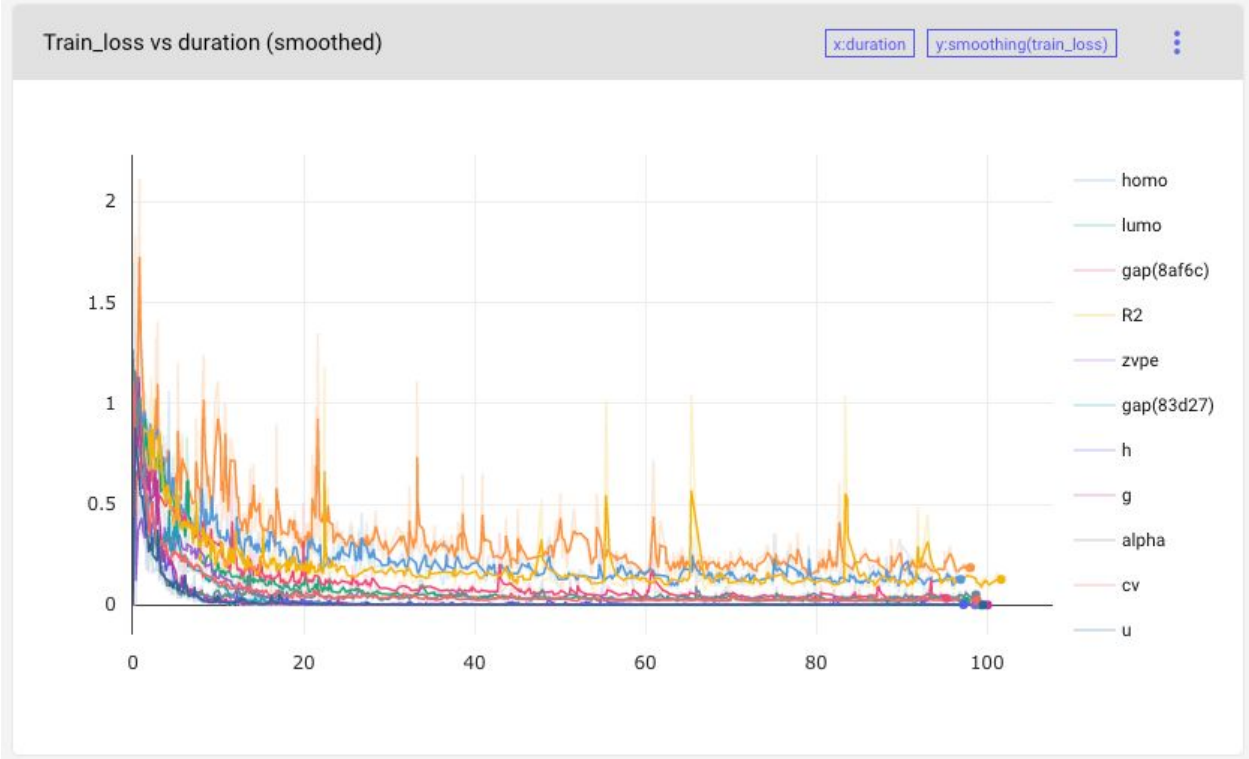
Starting learning rate: 0.001

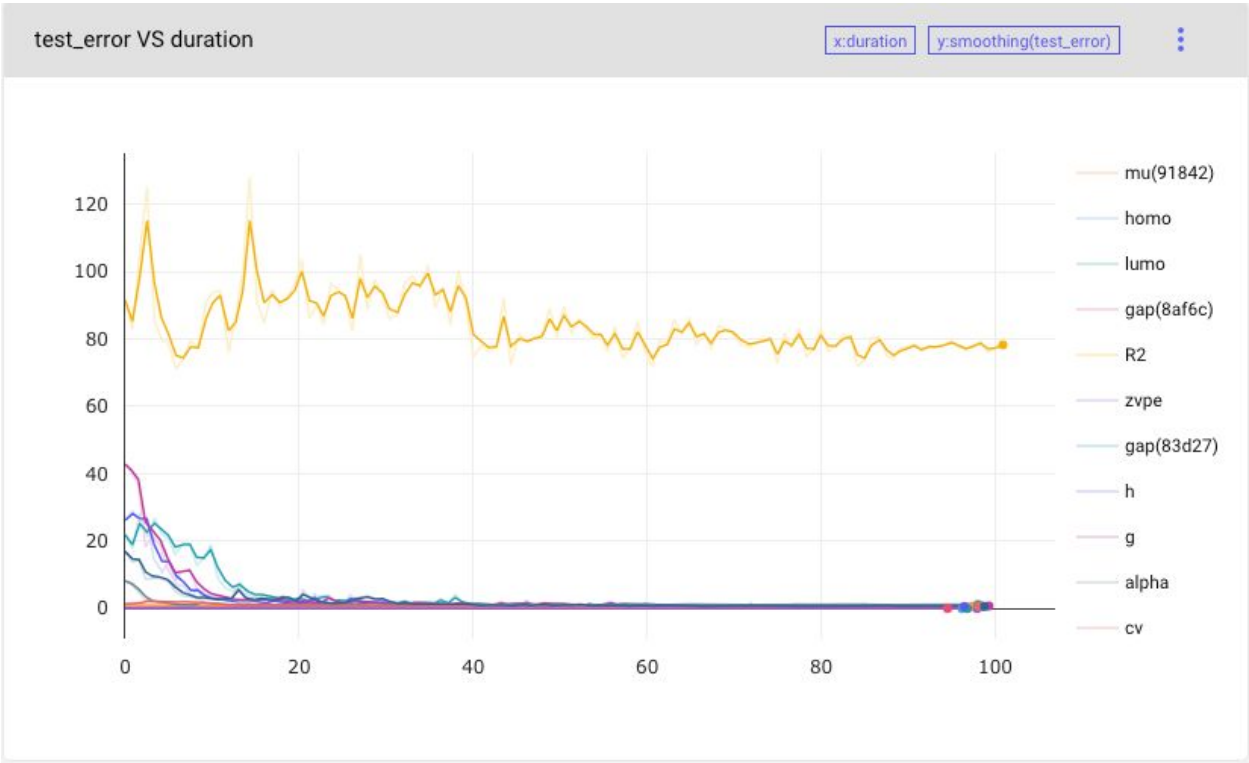
Training, validation and testing size: 11000, 1000,1000

More details later on...

SMILES + LSTM on target mu
https://www.comet.ml/bmbodj/smiles-descriptor/view/nAnxJP9sQAPQEbZ2j8Ub5Xyst




Train_loss vs duration (smoothed)
x:duration
y:smoothing(train_loss)
2
1.5
1
0.5
0
0
20
40
60
80
100
homo
lumo
gap(8af6c)
R2
zvpe
gap(83d27)
h
g
alpha
cv
u

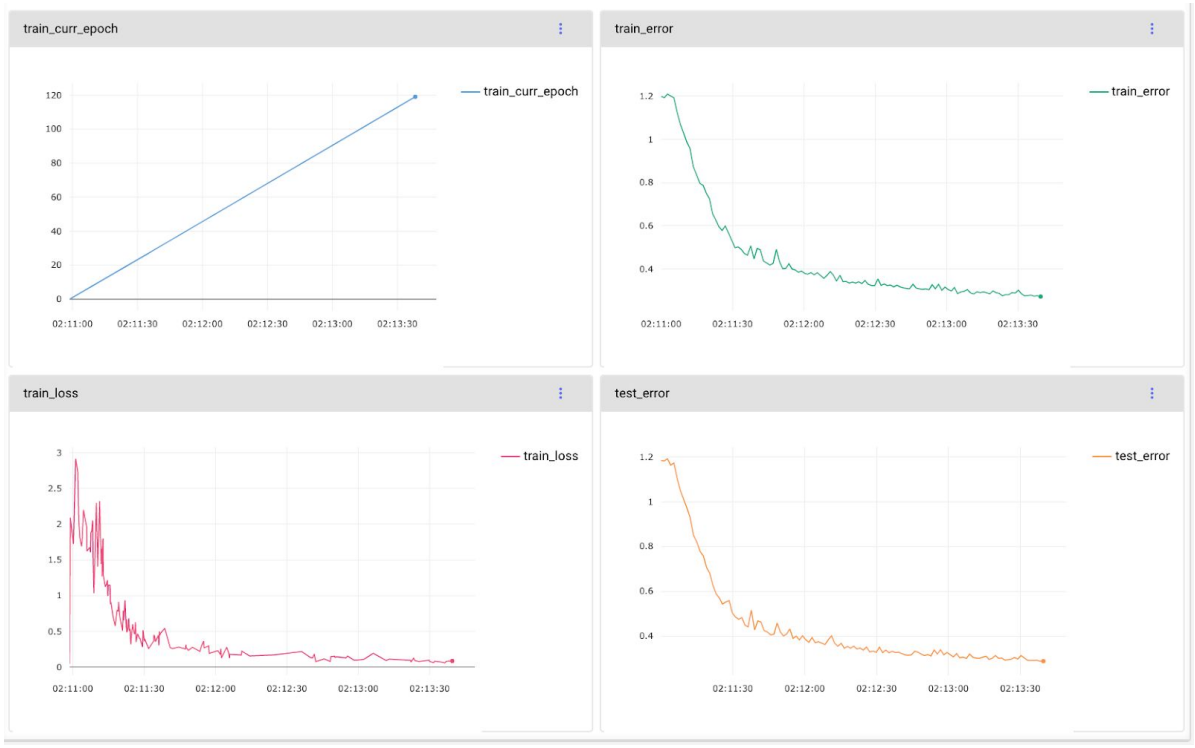

test_error VS duration
x:duration
y:smoothing(test_error)
120
100
80
60
40
20
0
0
20
40
60
80
100
mu(91842)
homo
lumo
gap(8af6c)
R2
zvpe
gap(83d27)
h
g
alpha
cv

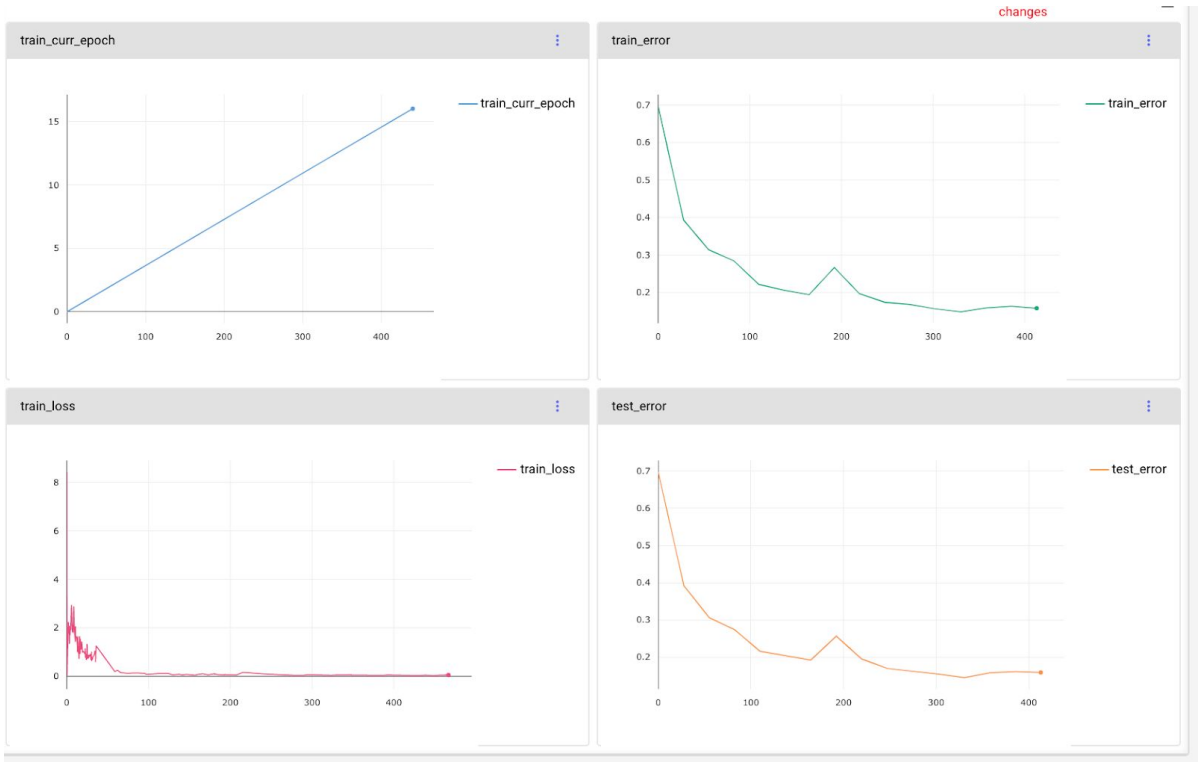
Name	Tags	Server ...	Duration	train_lo...	hidden...	train_c...	num_cl...	sequen...	test_er...	Batch ...	learnin...	num_e...
mu	mu	2/9/20 06:0...	00:17:37	0.1917638...	64	20460	1	25	0.6568424...	64	0.001	120
homo	homo	2/9/20 06:3...	00:10:58	0.1167926...	64	20460	1	25	0.0100424...	64	0.001	120
lumo	lumo	2/9/20 06:4...	00:09:36	0.0241778...	64	20460	1	25	0.0103132...	64	0.001	120
gap	gap	2/9/20 06:5...	00:10:22	0.0336005...	64	20460	1	25	0.0130798...	64	0.001	120
R2	R2	2/9/20 07:2...	00:32:57	0.1253733...	64	20460	1	25	78.851861...	64	0.001	120
zvpe	zpve	2/9/20 07:5...	00:25:17	0.0008285...	64	20460	1	25	0.0008520...	64	0.001	120
gap	u0	2/9/20 08:2...	00:12:10	0.0003496...	64	20460	1	25	0.7060622...	64	0.001	120
h	h	2/9/20 09:0...	00:04:30	0.0030353...	64	20460	1	25	0.6116014...	64	0.001	120
g	g	2/9/20 09:1...	00:08:30	0.0001425...	64	20460	1	25	0.8240498...	64	0.001	120
alpha	alpha	2/9/20 09:1...	00:09:53	0.0630368...	64	20460	1	25	1.2022591...	64	0.001	120
cv	cv	2/9/20 09:2...	00:10:31	0.0285819...	64	20460	1	25	0.6499725...	64	0.001	120
u	u	2/9/20 09:3...	00:12:23	0.0004261...	64	20460	1	25	0.6376352...	64	0.001	120

More Data ? (30k)

Name	Tags	Server ...	Duration	test_er...	File na...	hidden...	train_c...	train_lo...	sequen...	Batch ...	learnin...	num_e...
mu	mu	2/10/20 12:...	00:07:30	0.8441497...	Jupyter inte...	64	46520	0.1435420...	25	64	0.001	120
alpha	alpha	2/10/20 01:...	00:08:43	1.6165398...	Jupyter inte...	64	46520	0.0077846...	25	64	0.001	120
homo	homo	2/10/20 01:...	00:09:39	0.0095503...	Jupyter inte...	64	46520	0.0675823...	25	64	0.001	120
lumo	lumo	2/10/20 01:...	00:15:06	0.0125222...	Jupyter inte...	64	46520	0.0378192...	25	64	0.001	120
gap	gap	2/10/20 01:...	00:06:41	0.0178943...	Jupyter inte...	64	46520	0.0639942...	25	64	0.001	120
R2	r2	2/10/20 01:...	00:30:12	56.913784...	Jupyter inte...	64	46520	0.0161372...	25	64	0.001	120
zvpe	zvpe	2/10/20 02:...	01:05:58	0.0007383...	Jupyter inte...	64	46520	0.0008609...	25	64	0.001	120
u0	u0	2/10/20 12:...	00:56:58	0.5685452...	Jupyter inte...	64	31280	0.0001128...	25	64	0.001	120

Zinc dataset logP-13k and logp250k (up and down respectively)





Name	Tags	Serve...	File n...	Durati...	f	train_...	hidde...	train_...	num_...	sequ...	test_...	Batch...	learni...	num_...
qed-2...	qed-250k	2/10/20 ...	Jupyter i...	00:12:58	/root/.lo...	0.00246...	64	79810	1	110	0.04395...	64	0.001	120
logp-...	logp-250k	2/10/20 ...	Jupyter i...	00:09:02	/root/.lo...	0.05493...	64	59690	1	110	0.15874...	64	0.001	120
logp-...	logp-13k	2/10/20 ...	Jupyter i...	00:05:34	/root/.lo...	0.08375...	64	20460	1	110	0.28793...	64	0.001	120
qed-1...	qed-13k	2/10/20 ...	Jupyter i...	00:45:33	/root/.lo...	0.00667...	64	20460	1	110	0.06721...	64	0.001	120

Visualizations:

QM9-13k:https://www.comet.ml/bmbodj/smiles-descriptor/view/nAnxJP9sQAPQEbZ2j8Ub5Xyst

QM9-30k:https://www.comet.ml/bmbodj/smiles-descriptor30000/view/Rtk0EXlEQwCSsZdOzOHs47lzE

Zinc-13k&250k:https://www.comet.ml/bmbodj/smiles-descriptor-zinc/view/new

Sample Pytorch Implementation:

https://colab.research.google.com/drive/1H6p3xSwi6B-22PlKradOH3Scj8d9wml-

Size of largest molecule: 110 zinc, 26 QM9

Comparisons

Our results:

Target	Mu	Alpha	HOMO	LUMO	gap	R2	ZPVE	U0	U	H	G	cv
LSTM	0.65684	1.202	0.0100	0.01031	0.0131	78.8518	0.0008	0.70606	0.6116	0.82404	0.6499	0.6373

Baseline results:

Target	BOB	CM	GG-NN	MPNN
mu	0.6724	0.8078	0.7639	**0.1523**
alpha	0.7782	1.4809	0.9228	**0.3847**
HOMO	0.0074	0.017	0.0081	**0.0034**
LUMO	0.0106	0.0171	0.0111	**0.0037**
gap	0.0122	0.0196	0.0118	**0.0065**
R2	22.2674	37.7441	76.7278	**2.5781**
ZPVE	0.0007	0.0010	0.0007	**0.0004**
U0	0.7360	3.4899	**0.3955**	0.5545
U	0.7360	3.4895	**0.2621**	0.5218
H	0.7360	3.4898	**0.3899**	0.3991
G	0.7360	3.4901	**0.4603**	0.4632
CV	0.4156	0.6600	0.5121	**0.1516**

ChemVAE results:

Database/Property	Mean[a]	ECFP[b]	CM[b]	GC[b]	1-hot SMILES[c]	Encoder[d]	VAE[e]
ZINC250k/logP	1.14	0.38	-	0.05	0.16	0.13	0.15
ZINC250k/QED	0.112	0.045	-	0.017	0.041	0.037	0.054
QM9/HOMO, eV	0.44	0.20	0.16	0.12	0.12	0.13	0.16
QM9/LUMO, eV	1.05	0.20	0.16	0.15	0.11	0.14	0.16
QM9/Gap, eV	1.07	0.30	0.24	0.18	0.16	0.18	0.21

Our results:

Size/property	LogP	QED
ZINC-13K	0.28793	0.067213
ZINC-250K	0.158749	0.0439576

Questions/ (continued)

-https://github.com/bmbodj/COMP396/blob/master/Fall_2019/COMP396_report.pdf

- Is the edge network/ or any graph model that handles edge attributes a must for efficient quantum property prediction tasks on molecules? (OR) Would it depend on the task (node, graph or edge, level)?

-Which graph model would you consider as the state of the art for quantum property predictions?

I found a paper that uses DGI to maximize information between edge states and transform parameters.-> Utilizing Edge Features in Graph Neural Networks

via Variational Information Maximization https://arxiv.org/pdf/1906.05488.pdf

Next Steps?

- You tell me :)

- read the paper on the semi-supervised application of DGI
https://arxiv.org/pdf/1908.01000.pdf

- get an in-depth understanding of DGI

- Think about how to implement DGI to match LSTM and GNN molecular representations