Practice 05

KAIST CHEMISTRY

SMILES and CNN

Seongok Ryu

Department of Chemistry, KAIST

KAIST

Contents

- End-to-end learning
- SMILES – molecular representation by a string
- Prediction of logP using SMILES and CNN
- Assignment #4

How can we improve the models? (at the last lecture)

Possibilities

- **Learning rate is too small or big.**
- **Missing regularizations (prior regularization, dropout)**
- **Input, the molecular fingerprint, is not good.**
- **Need better model, instead of MLP**

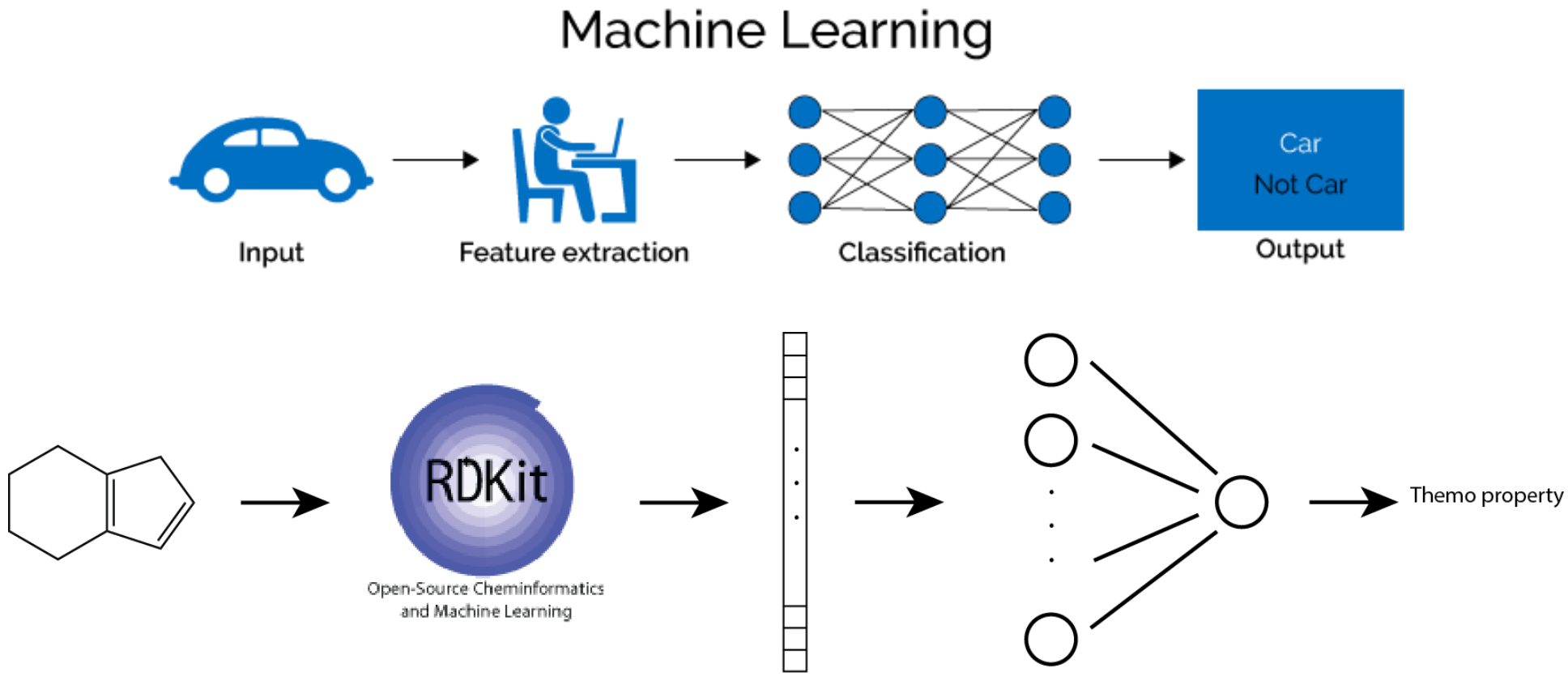
→ Using raw input, e.g.) SMILES, molecular graph

rather than featurized inputs, e.g.) molecular fingerprint

→ Using better model, e.g.) CNN, RNN, Graph NN.

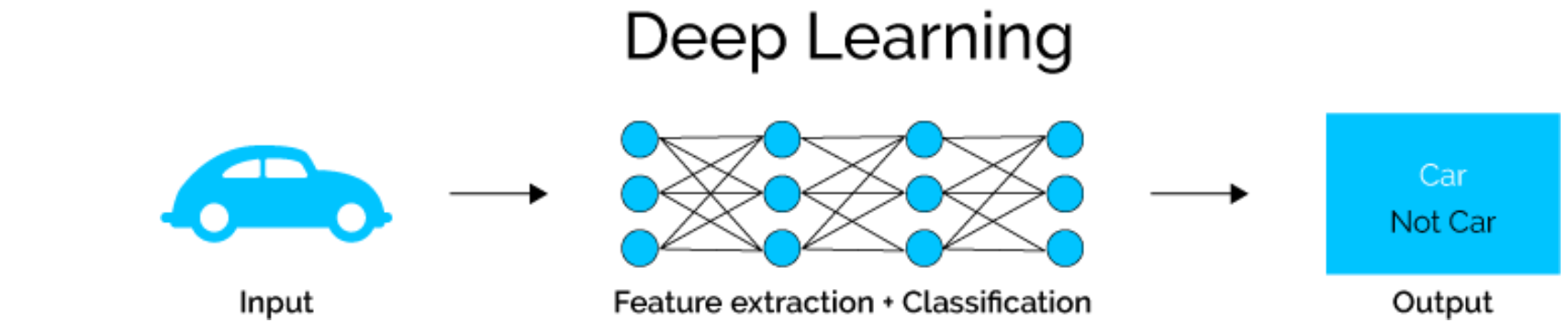
End-to-end learning

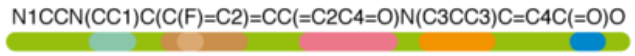
- We used molecular fingerprints for inputs of the models.
- The molecular fingerprint is a “featurized(pre-processed) input.
- It means that an expert(prior)-knowledge can be unintentional biases.

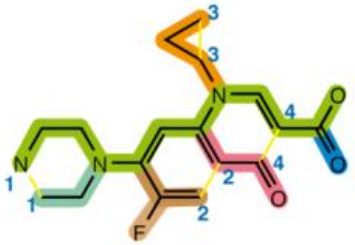


End-to-end learning

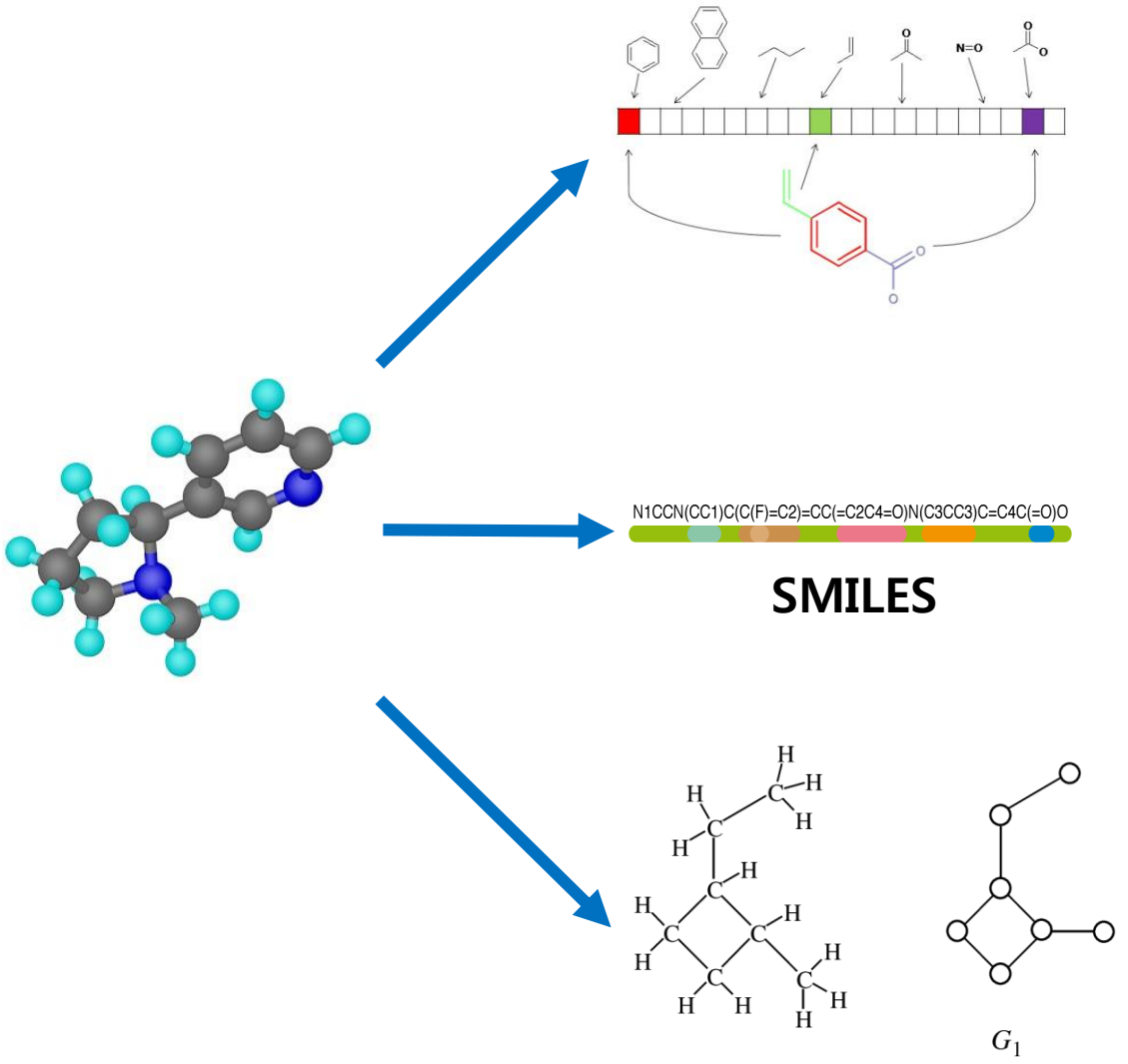
- How about use raw inputs rather than featurized inputs?
- SMILES and molecular graph can describe the molecular structure.
- Let machines to learn both featurization and prediction by itself. – It is the heart of deep learning!







SMILES – molecular representation by a string



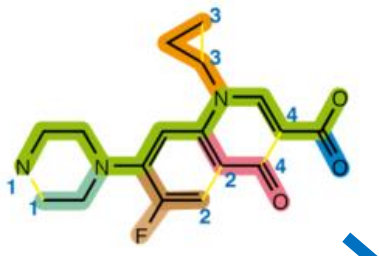
KAIST

SMILES – molecular representation by a string

*"The **simplified molecular-input line-entry system** (**SMILES**) is a specification in form of a line notation for describing the structure of chemical species using short ASCII strings. SMILES strings can be imported by most molecule editors for conversion back into two-dimensional drawings or three-dimensional models of the molecules."*

https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system

Prediction of logP using SMILES and CNN



N	1	C	C	N	(	C	C	1	)	C	(	C	(	F	)	=	C	2	)	=	C	C	(	=	C	2	C	4	=	O	)	N	(	C	3	C	C	3	)	C	=	C	4	C	(	=	O	)	O

shape = [#batch, #characters]

Prediction of logP using SMILES and CNN

N	1	C	C	N	(	C	C	1	)	C	(	C	(	F	)	=	C	2	)	=	C	C	(	=	C	2	C	4	=	O	)	N	(	C	3	C	C	3	)	C	=	C	4	C	(	=	O	)	O

One-hot encoding

C	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0
N	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
…	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

KAIST

shape = [#batch, #characters, #types]

Prediction of logP using SMILES and CNN

C	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0
N	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1d convolution

KAIST

Prediction of logP using SMILES and CNN

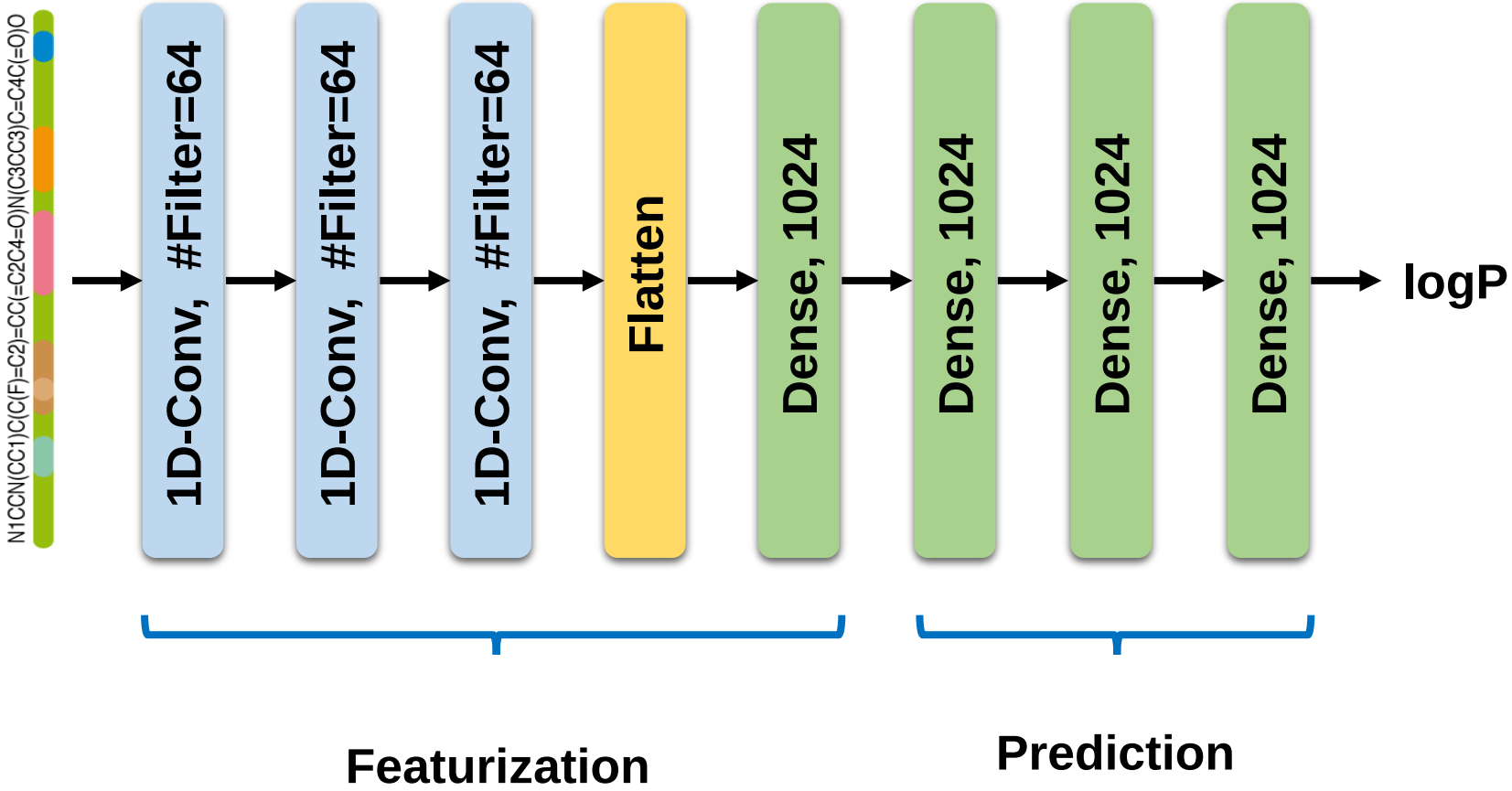
C	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	0	0
N	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
=	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1d convolution, stride=1, padding='same'

output shape = [#batch, #characters, #filters]

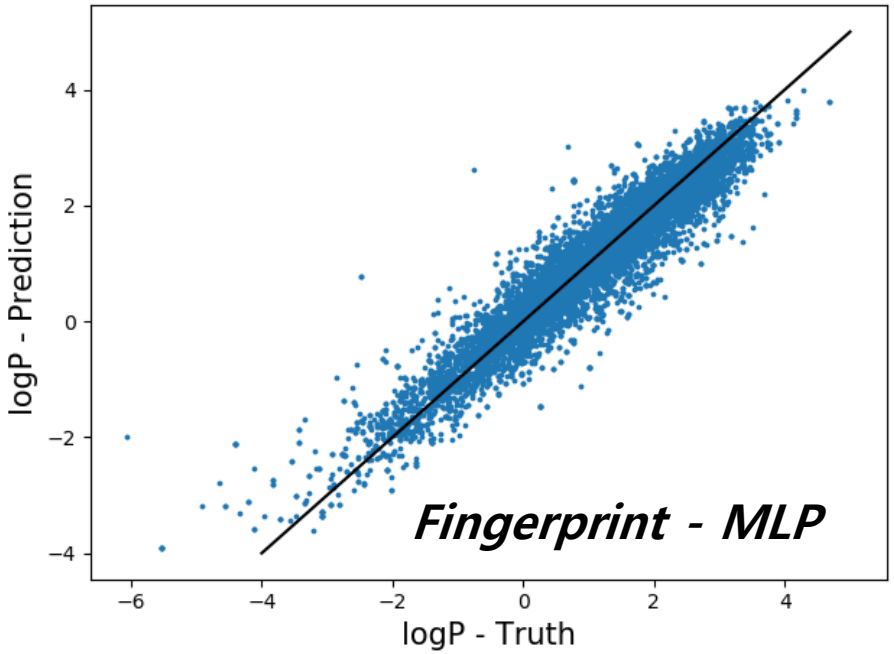
Prediction of logP using SMILES and CNN

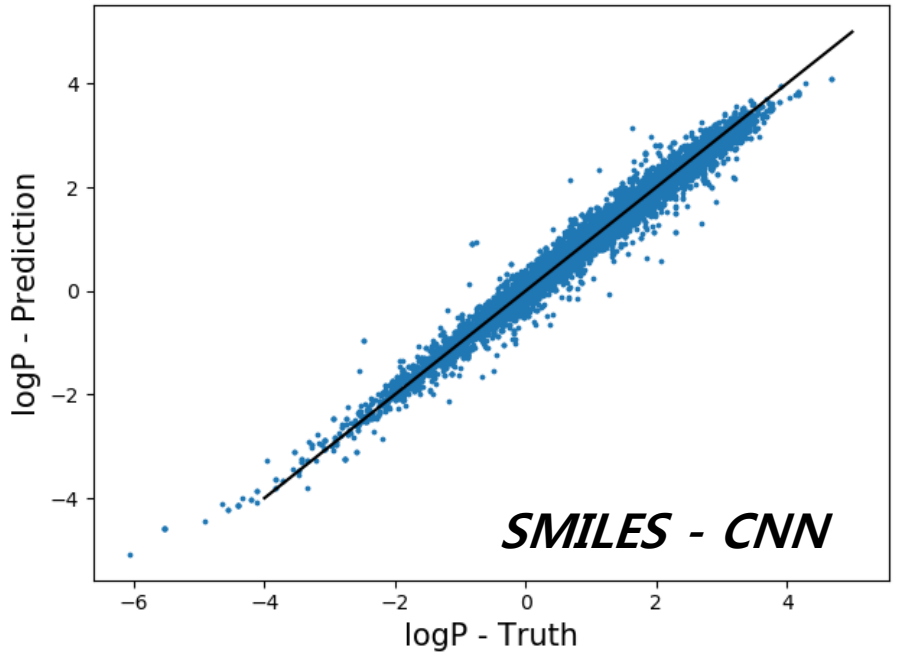
Overall architecture of the CNN model



Prediction of logP using SMILES and CNN

Results





- ✓ Hidden dim = 1024
- ✓ Regularization lambda = 0.01
- ✓ No dropout
- ✓ 30000/1000/1000 – train/val/test

	Fingerprint - MLP	SMILES - CNN	Graph - later
MAE	**0.31**	**0.42**	**?**
Std. dev	**0.15**	**0.20**	**?**

KAIST

Assignment #4

Improve the vanilla CNN model

- In this class, TA showed vanilla the CNN model – which is consisted of three convolutional layers and predictor composed of three dense layers.
- We learned wider and deeper model – InceptionNet and ResNet in today's lecture.
- Therefore, try to develop better model than the vanilla CNN.
  Using the inception modules or deeper model with the skip-connections can improve the vanilla CNN.
- Report your results - MAE, std. dev, and truth-prediction plot.