Practice 08

Overview on molecular property predictions

Seongok Ryu

Department of Chemistry, KAIST

Codes available with scripts at https://github.com/SeongokRyu/augmented-GCN

Contents

- Overview on molecular property predictions
- Message passing neural network
- Assignment #7

Overview on molecular property predictions

Overview

What we have done?

→ Supervised learning of molecular properties, e.g. logP and toxicity

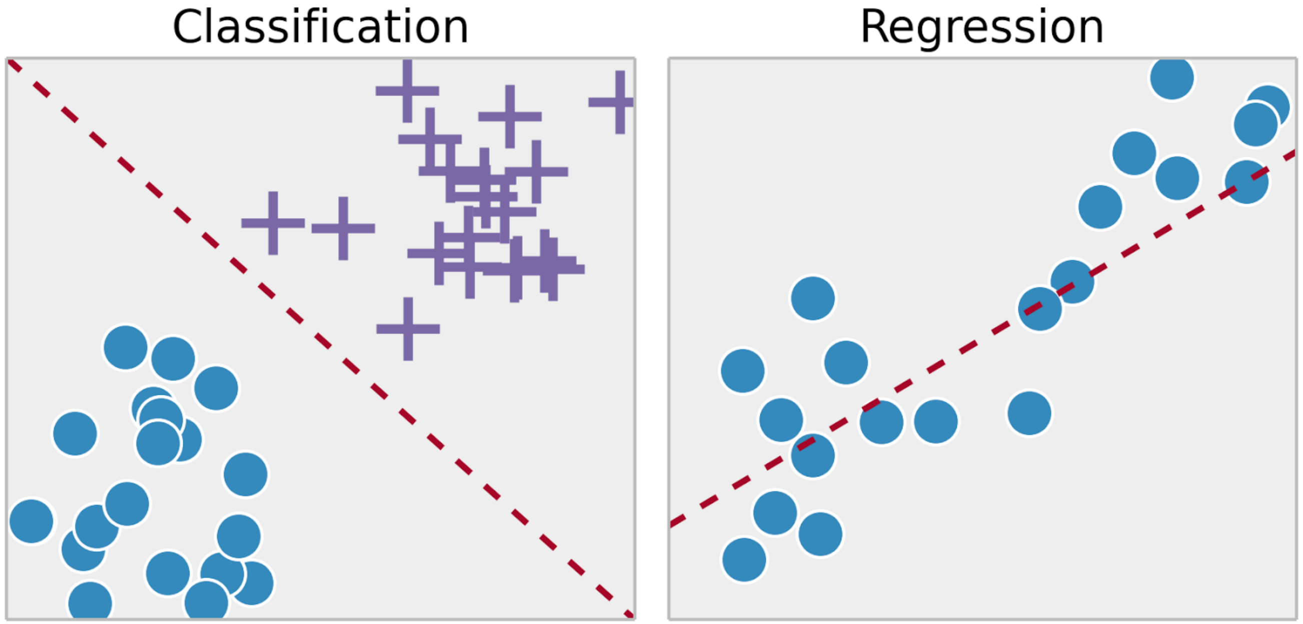
Overview

What we have done?

→ Supervised learning of molecular properties, e.g. logP and toxicity

What is supervised learning?

→ Learning a function that maps an input X to an output Y based on given $\{X, Y\}$ pairs



Overview

What we have done?

→ Supervised learning of molecular properties, e.g. logP and toxicity

What is supervised learning?

→ Learning a function that maps an input X to an output Y based on given $\{X, Y\}$ pairs

For what?

→ Statistical inference: the process of using data analysis to deduce properties of an underlying probability distribution

$$p(\boldsymbol{\theta}|\mathbf{X},\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X},\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{Y}|\mathbf{X})}$$

Overview

How can we obtain nice models?

→ Using proper dataset, $\{X, Y\}$

: molecular fingerprint (structural descriptor), SMILES, molecular graph

: qualified dataset – amount and quality

Overview

How can we obtain nice models?

→ Using proper dataset, $\{X, Y\}$

: molecular fingerprint (structural descriptor), SMILES, molecular graph

: qualified dataset – amount and quality

→ Setting the appropriate model

: linear regression, support vector machine (SVM)

: multi-layer perceptron (MLP), convolutional neural network (CNN),
recurrent neural network (RNN), graph neural network (GNN)

Overview

How can we obtain nice models?

→ Using proper dataset, $\{X, Y\}$

: molecular fingerprint (structural descriptor), SMILES, molecular graph

: qualified dataset – amount and quality

→ Setting the appropriate model

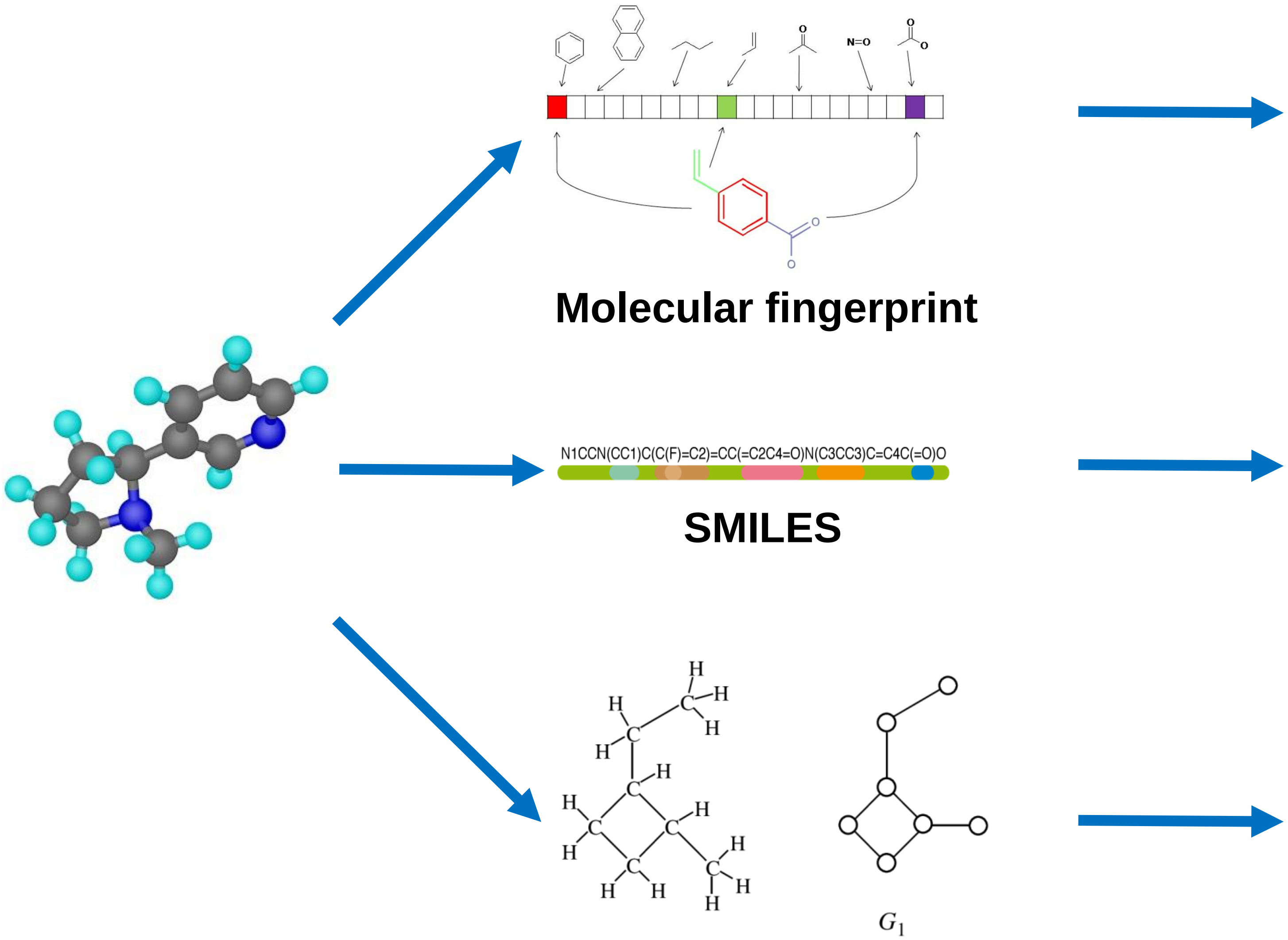
: linear regression, support vector machine (SVM)

: multi-layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), graph neural network (GNN)

→ Avoid overfitting

: controlling model capacity, regularization

Overview



Molecular fingerprint

SMILES

Molecular graph

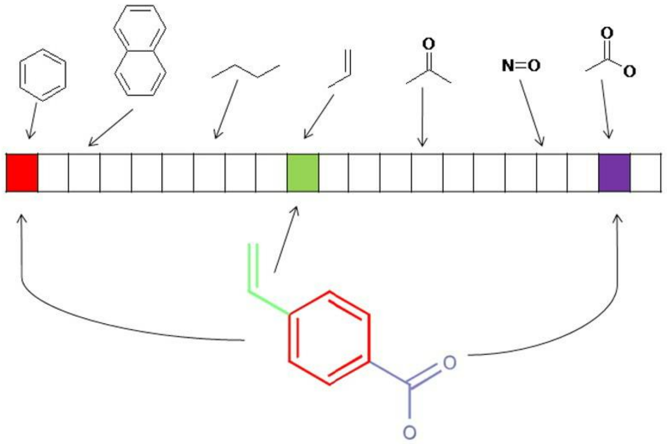
SVM – practice 03
MLP – practice 04

CNN – practice 05
RNN – practice 07

GCN – practice 06
MPNN(GGNN) – practice 08

Overview

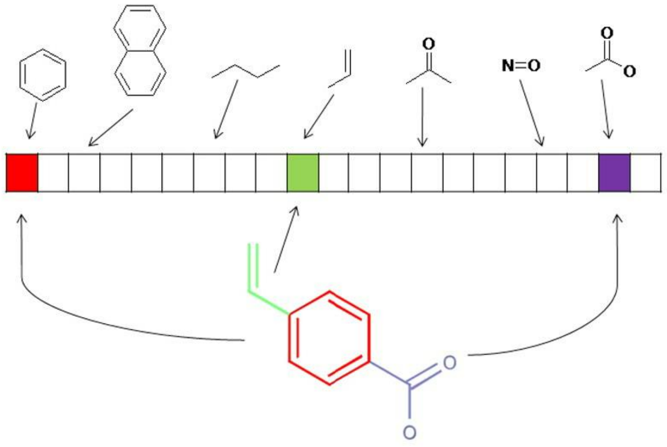
Molecular fingerprint



- ✓ Structural descriptor featurized by the deterministic algorithm, i.e.) hash function
- ✓ Do not require additional parameters to featurize molecules
- ✓ However, it can miss information in featurization process

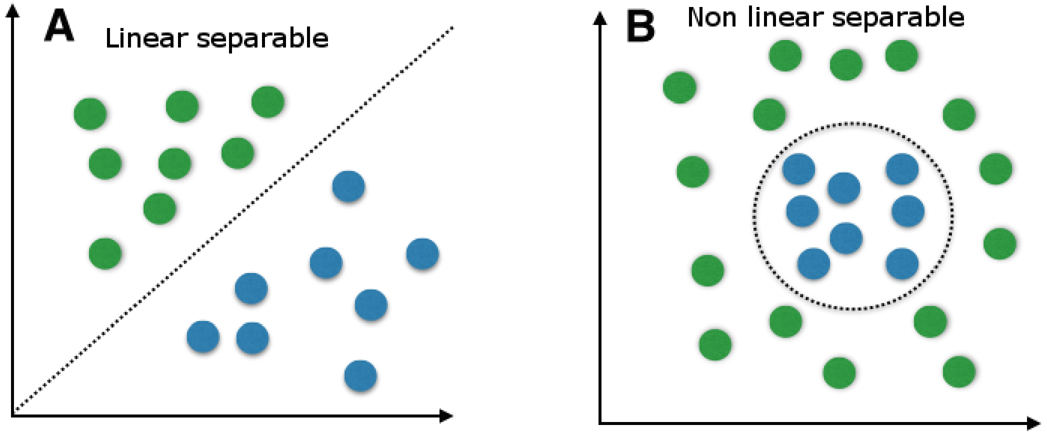
Overview

Molecular fingerprint



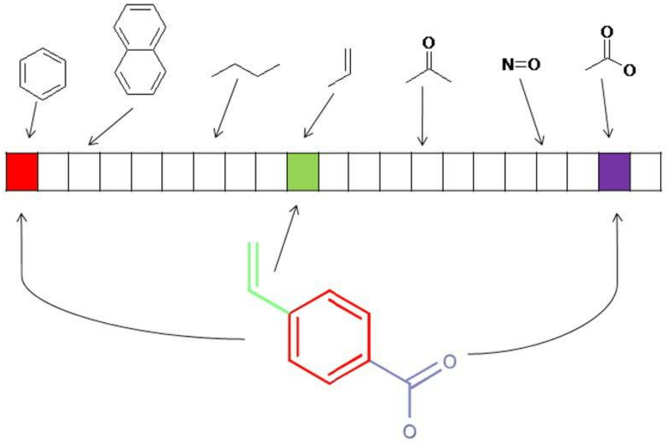
- ✓ Structural descriptor featurized by the deterministic algorithm, i.e.) hash function
- ✓ Do not require additional parameters to featurize molecules
- ✓ However, it can miss information in featurization process

SVM : relatively low number of parameters



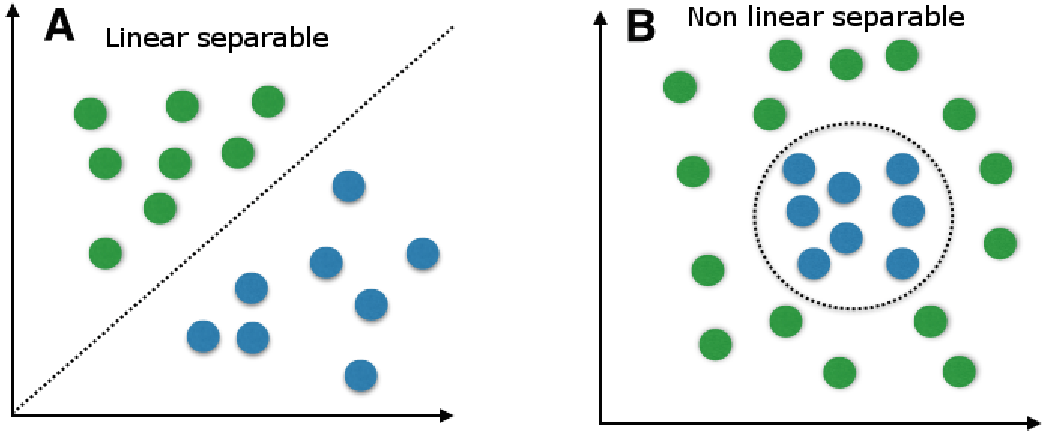
Overview

Molecular fingerprint

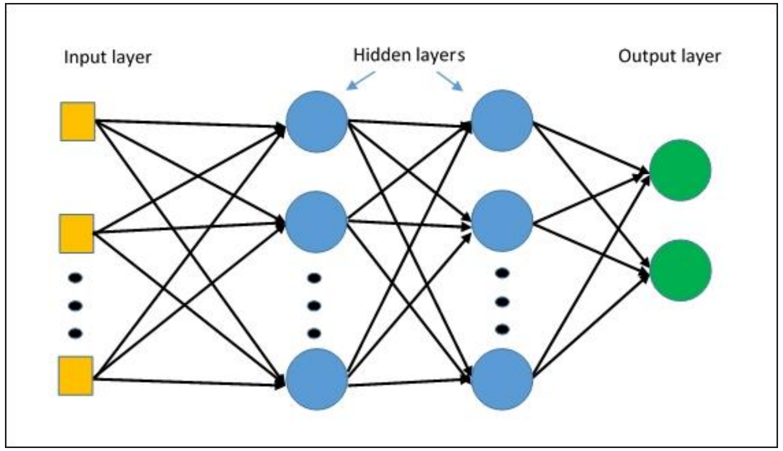


- ✓ Structural descriptor featurized by the deterministic algorithm, i.e.) hash function
- ✓ Do not require additional parameters to featurize molecules
- ✓ However, it can miss information in featurization process

SVM : relatively low number of parameters

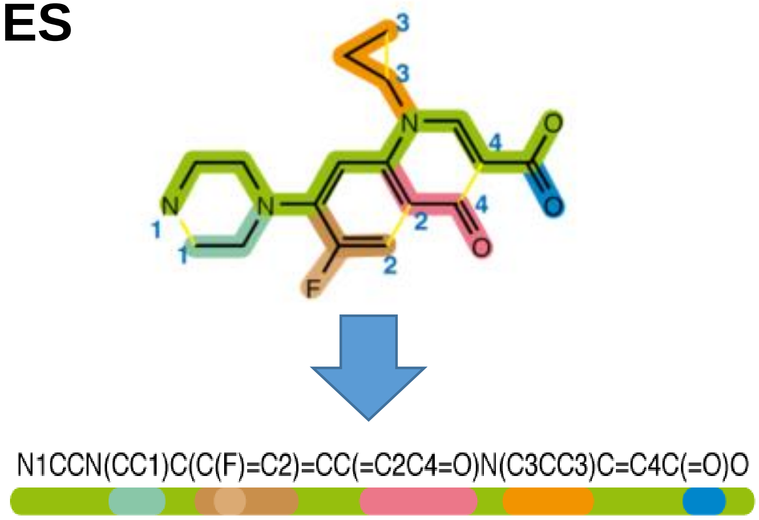


MLP : large number of parameters
universal function approximator



Overview

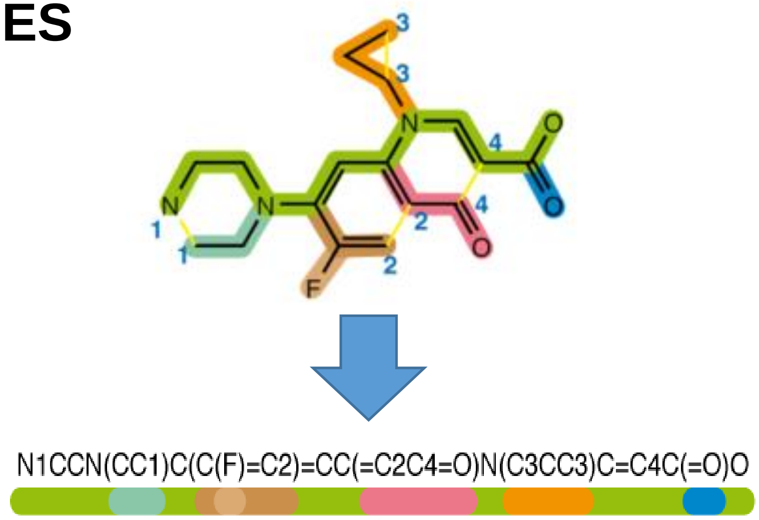
SMILES



- ✓ String representation of molecular structure
- ✓ Most of small drug molecules can be represented with less than 120 characters
- ✓ Useful for digitizing moleucles
- ✓ However, its topological information can be spoiled.

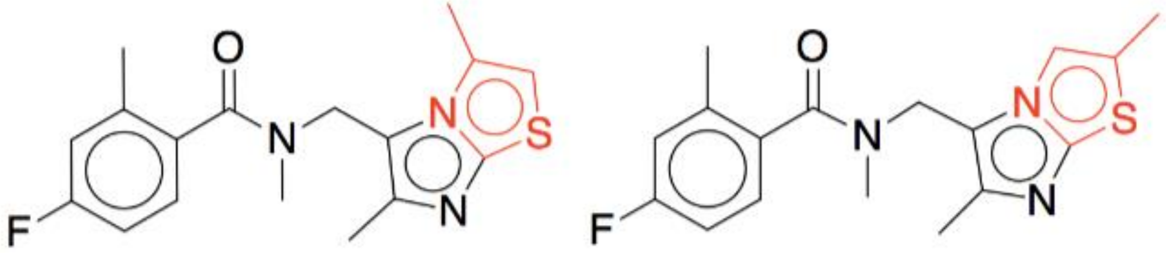
Overview

SMILES



N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

- ✓ String representation of molecular structure
- ✓ Most of small drug molecules can be represented with less than 120 characters
- ✓ Useful for digitizing moleucles
- ✓ However, its topological information can be spoiled.



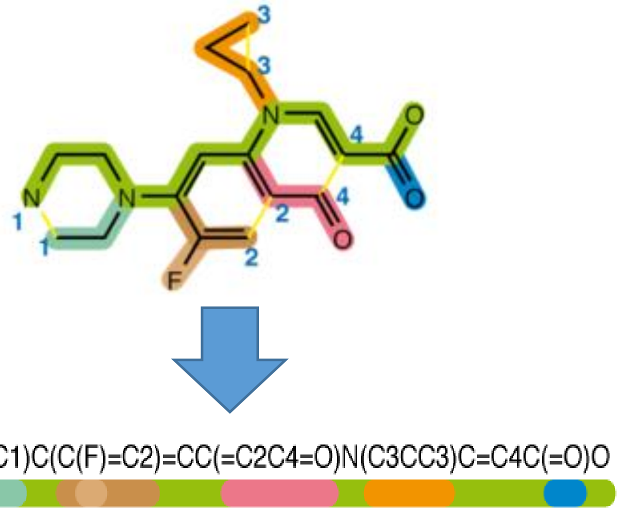
Cc1cn2c(CN(C)C(=O)c3ccc(F)cc3C)c(C)nc2s1

Cc1cc(F)ccc1C(=O)N(C)Cc1c(C)nc2scc(C)n12

Figure 1. Two almost identical molecules with markedly different canonical SMILES in RDKit. The edit distance between two strings is 22 (50.5% of the whole sequence).

Overview

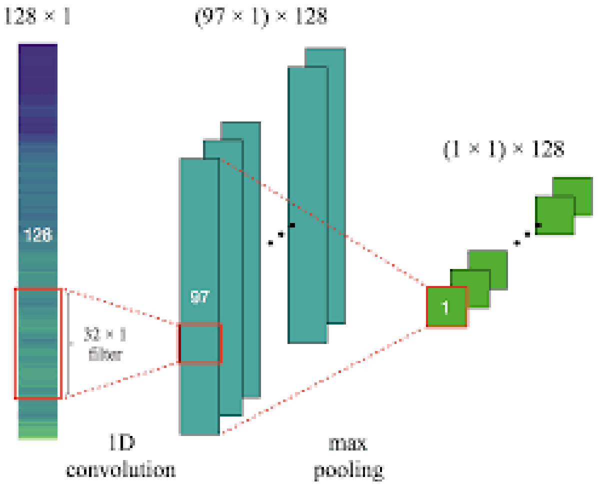
SMILES



N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

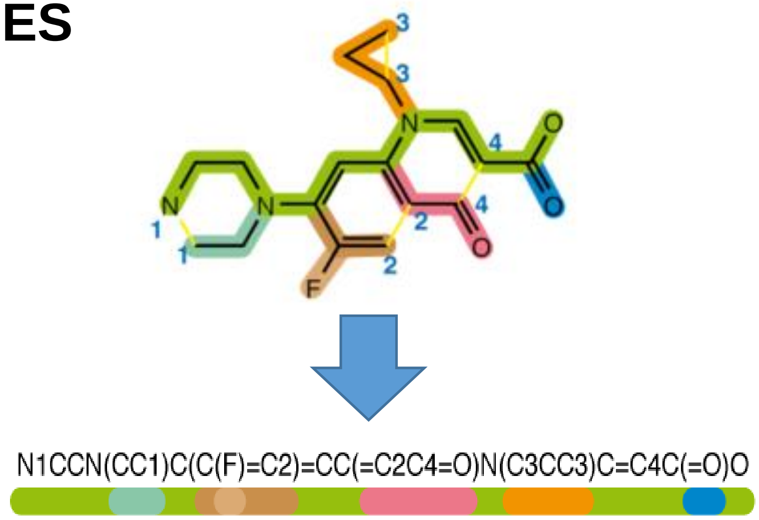
- ✓ String representation of molecular structure
- ✓ Most of small drug molecules can be represented with less than 120 characters
- ✓ Useful for digitizing moleucles
- ✓ However, its topological information can be spoiled.

CNN : sampling entities effectively
with parameter sharing



Overview

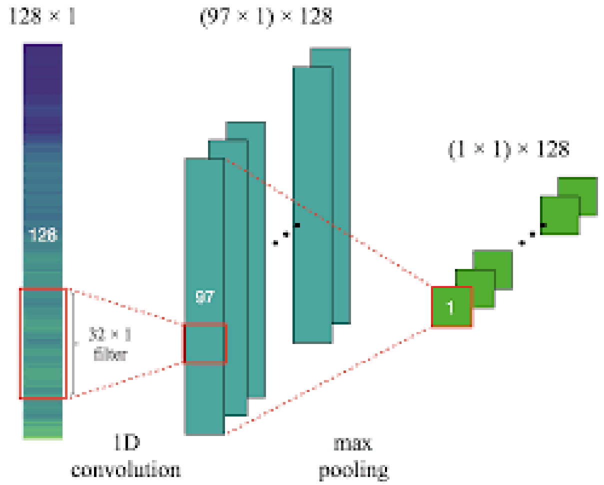
SMILES



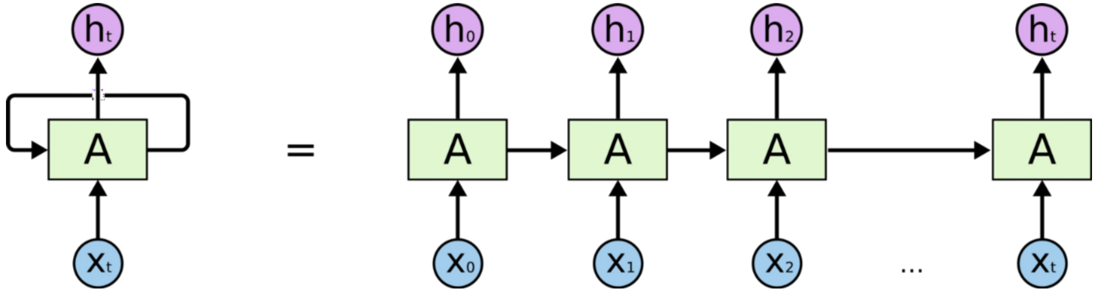
N1CCN(CC1)C(C(F)=C2)=CC(=C2C4=O)N(C3CC3)C=C4C(=O)O

- ✓ String representation of molecular structure
- ✓ Most of small drug molecules can be represented with less than 120 characters
- ✓ Useful for digitizing moleucles
- ✓ However, its topological information can be spoiled.

CNN : sampling entities effectively with parameter sharing

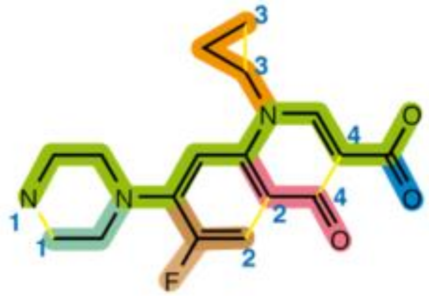


RNN : find relations between entities effectively



Overview

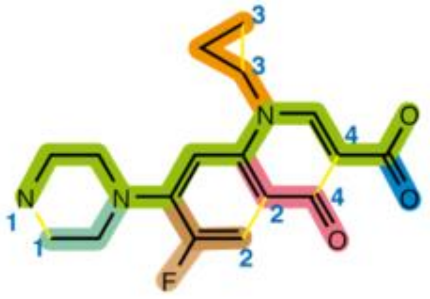
Molecular graph



- ✓ 2D representation of molecular structures
- ✓ Nodes : atom descriptors, e.g.) atom types, # of hydrogen
- ✓ Edges : connectivity between atoms, bond types, distance
- ✓ Most powerful except conformational information is required

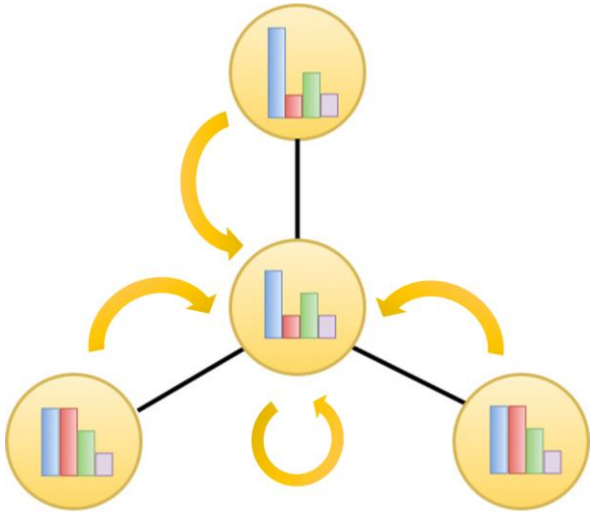
Overview

Molecular graph



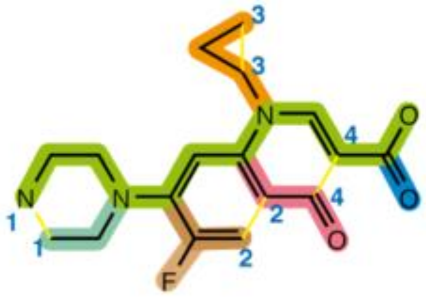
- ✓ 2D representation of molecular structures
- ✓ Nodes : atom descriptors, e.g.) atom types, # of hydrogen
- ✓ Edges : connectivity between atoms, bond types, distance
- ✓ Most powerful except conformational information is required

GCN : conv-net suitable for graph structure



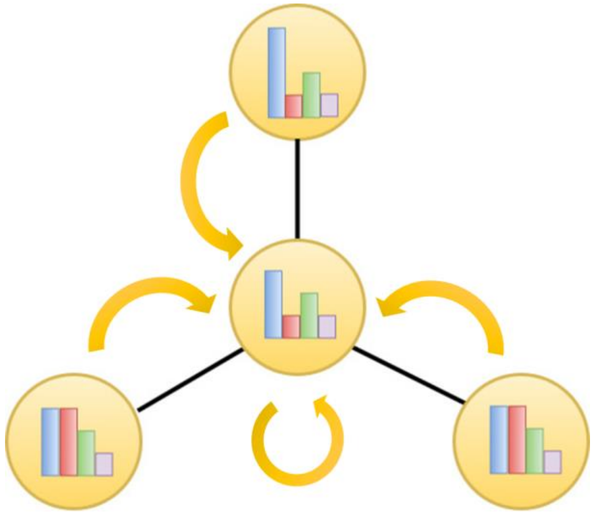
Overview

Molecular graph

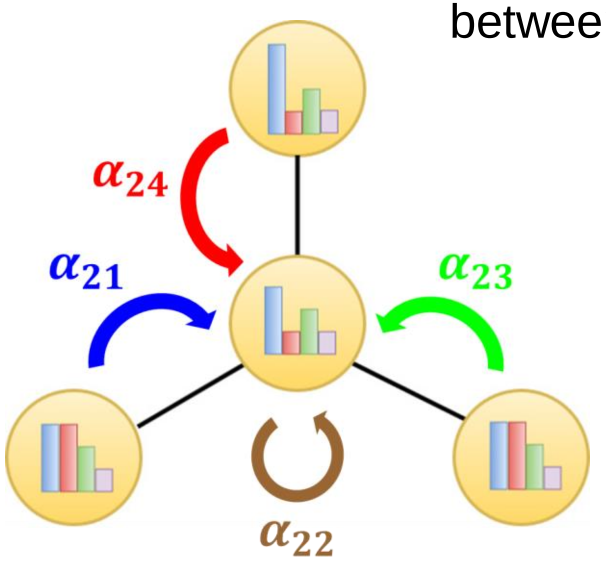


- ✓ 2D representation of molecular structures
- ✓ Nodes : atom descriptors, e.g.) atom types, # of hydrogen
- ✓ Edges : connectivity between atoms, bond types, distance
- ✓ Most powerful except conformational information is required

GCN : conv-net suitable for graph structure



GAT : applying attention to capture relations between atoms



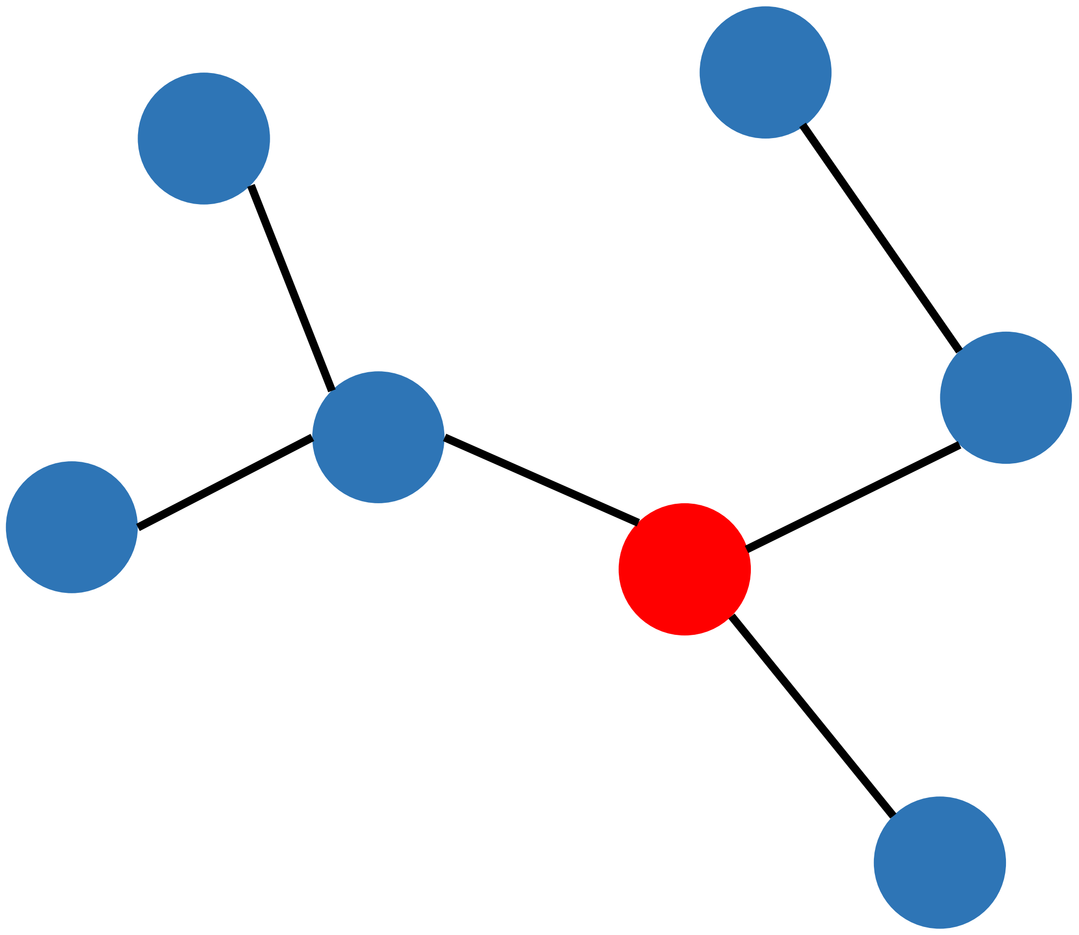
Accuracy of different models

Accuracy in logP predictions

	FP - MLP	SMILES – CNN	Graph – GCN	SMILES – RNN1	SMILES – RNN2	SMILES – RNN3	…	My best (???)
MAE	0.31	0.15	0.088	0.13	0.05	0.072	…	0.01
Std.dev	0.42	0.20	0.137	0.18	0.08	0.11	…	-

Message passing neural network

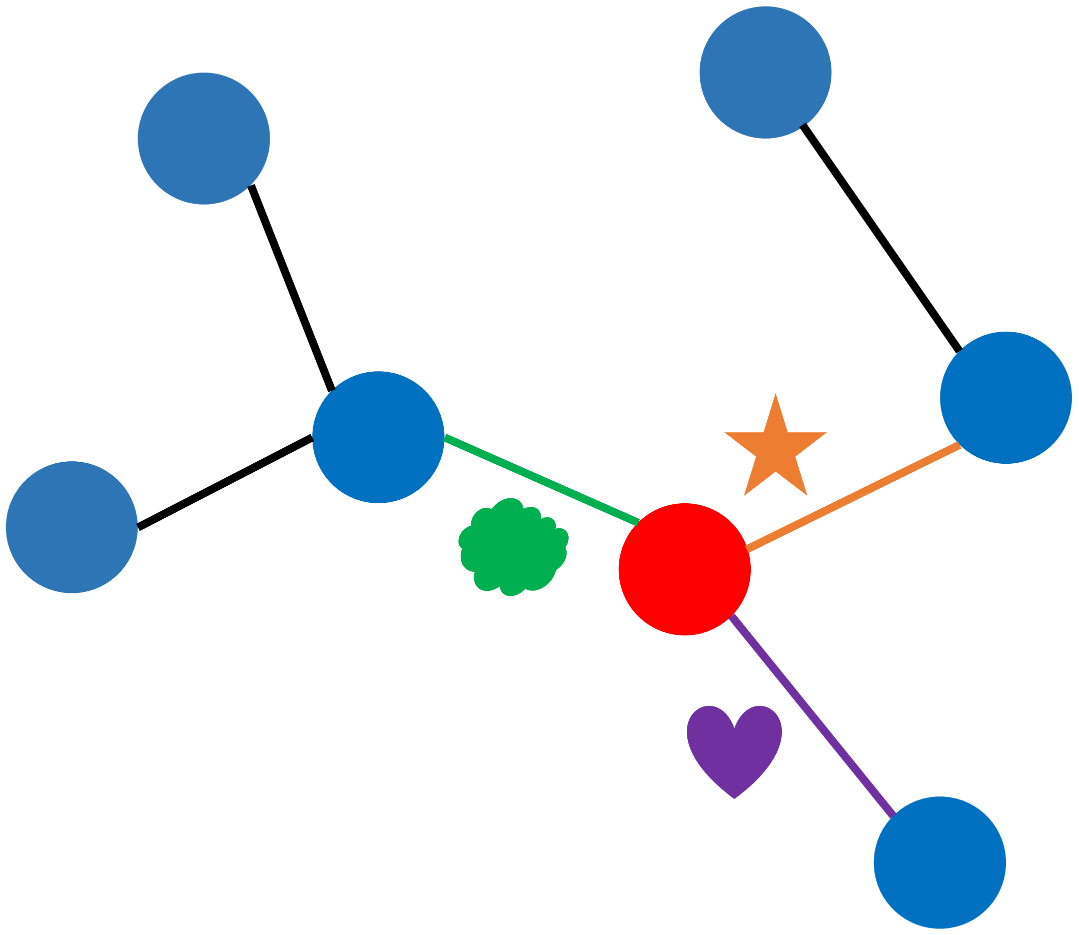
Message passing neural network



When we update the i-th node state 🔴

MPNN : $H_i^{(l)} = f(H_i^{(l)}, m_i^{(l+1)})$

Message passing neural network



When we update the i-th node state 🔴

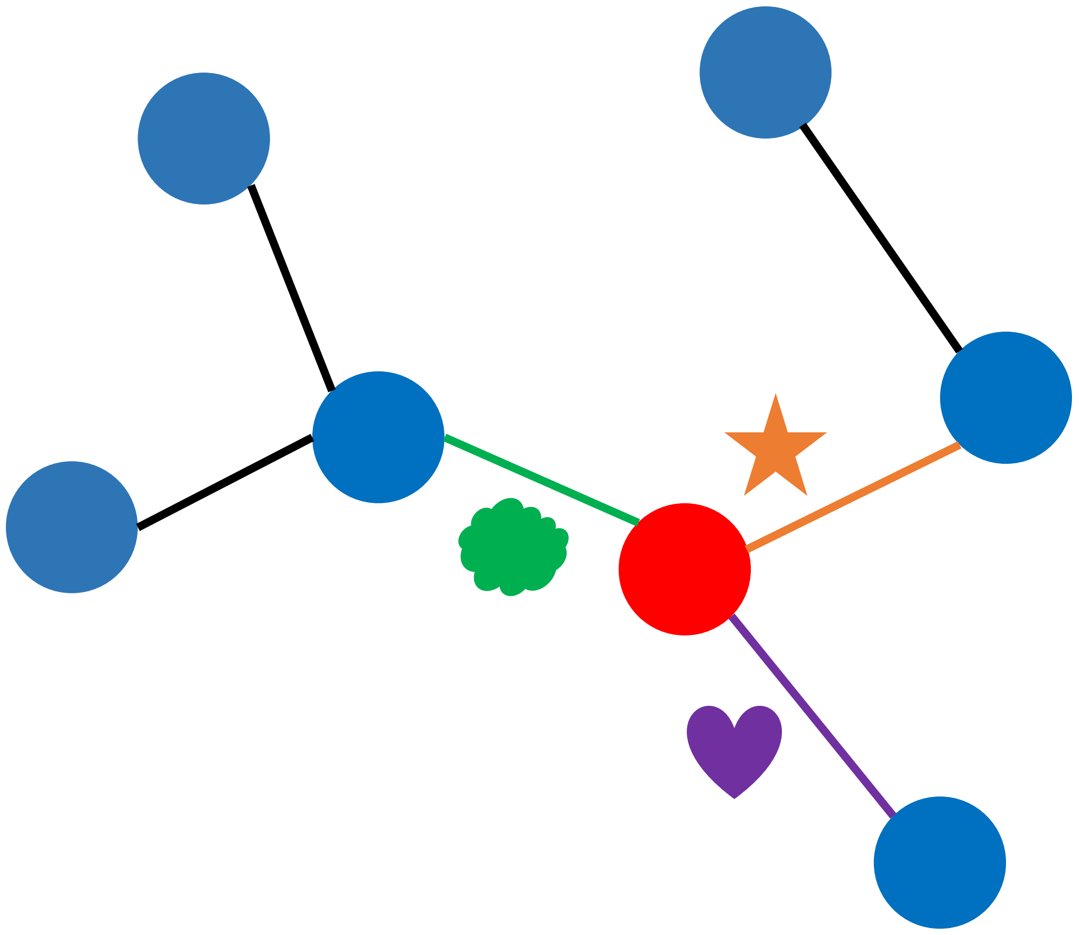
MPNN : $H_i^{(l)} = f(H_i^{(l)}, m_i^{(l+1)})$

$m_{ij}^{(l+1)} =$ ☁ $= f($ 🔵 , 🔴 , ━ $)$

The message state $m_{ij}^{(l+1)}$ is updated as a function of i- and j-th node states and edge features.

$m_i^{(l+1)} =$ [★ + ☁ + ♥]

Message passing neural network



When we update the i-th node state ●

MPNN : $H_i^{(l)} = f(H_i^{(l)}, m_i^{(l+1)})$

$$m_{ij}^{(l+1)} = M^{(l)}\left(H_i^{(l)}, H_j^{(l)}, e_{ij}\right)$$

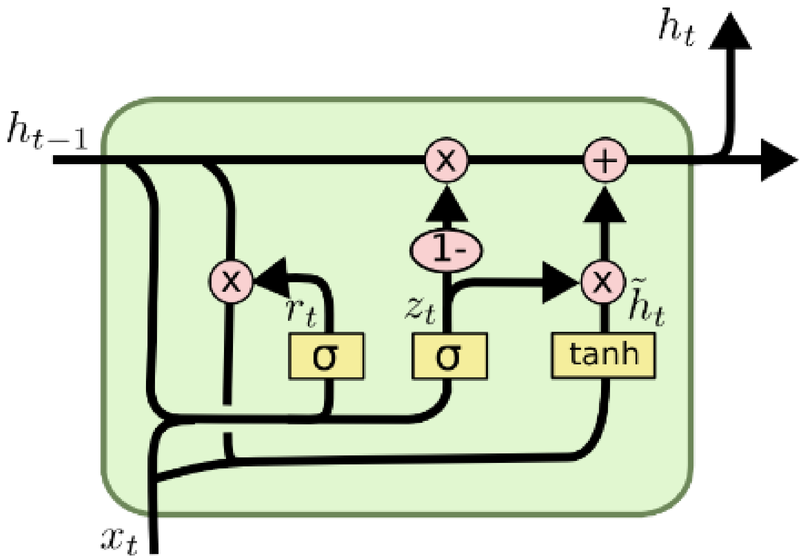
e_{ij} : ex) single/double/aromatic/… bond

$$m_i^{(l+1)} = \sum_{j \in N_i} m_{ij}^{(l+1)}$$

$$H_i^{(l)} = \mathrm{GRU}(H_i^{(l)}, m_i^{(l+1)})$$

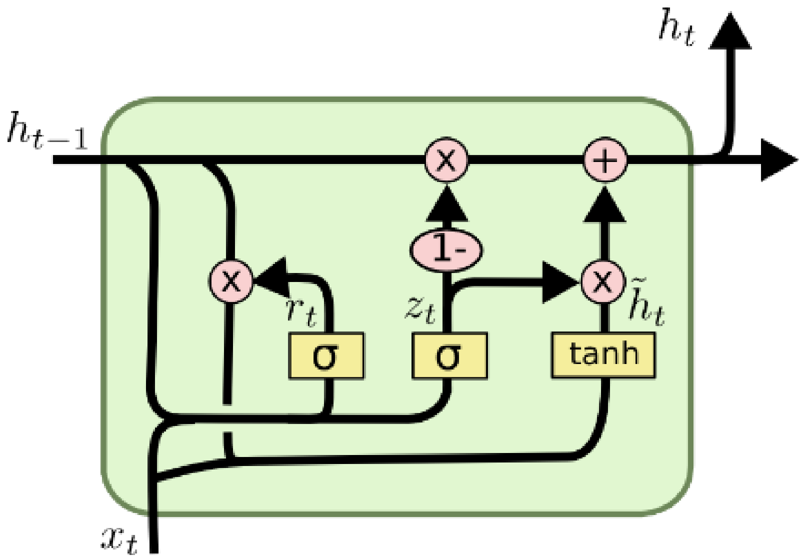
Message passing neural network

GGNN : using GRU for updating and message states as summation of adjacent node states.



Message passing neural network

GGNN : using GRU for updating and message states as summation of adjacent node states.



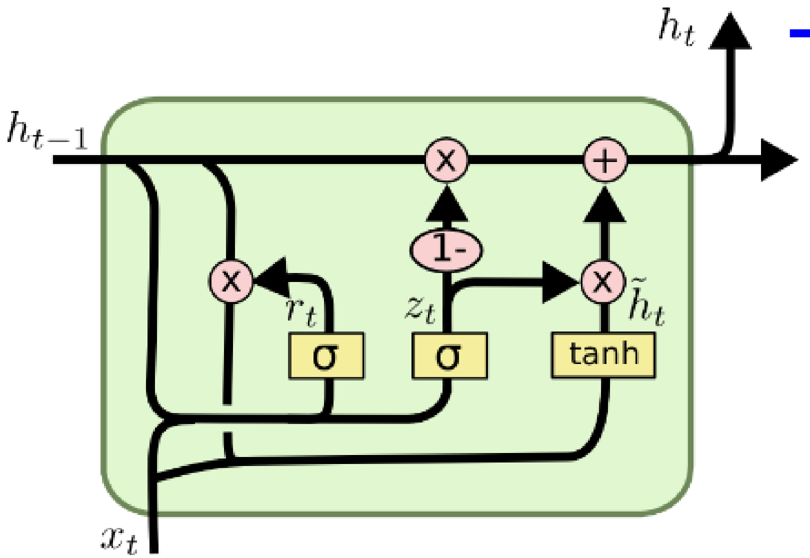
In this case, $x_t = m_i^{(l+1)}$

$$m_i^{(l+1)} = \sum_j H_j^{(l)}$$

Codes with scripts at https://github.com/SeongokRyu/augmented-GCN

Message passing neural network

GGNN : using GRU for updating and message states as summation of adjacent node states.



$$H_i^{(l+1)} = \left(1 - z_i^{(l)}\right) \odot H_i^{(l)} + z_i^{(l)} \odot \tilde{H}_i^{(l+1)}$$

: weighted summation with update rate $z_i^{(l)}$ of temporary state $\tilde{H}_i^{(l+1)}$ and previous state $H_i^{(l)}$

In this case, $x_t = m_i^{(l+1)}$

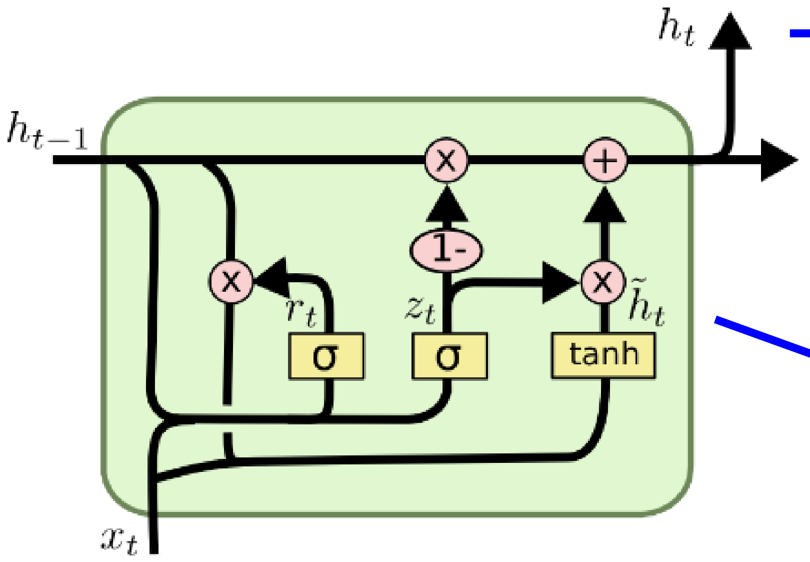
$$m_i^{(l+1)} = \sum_j H_j^{(l)}$$

$r_i^{(l+1)} = \sigma\left(W_r \cdot \left[H_i^{(l+1)}, m_i^{(l+1)}\right]\right)$: forget rate

Codes with scripts at https://github.com/SeongokRyu/augmented-GCN

Message passing neural network

GGNN : using GRU for updating and message states as summation of adjacent node states.



$$H_i^{(l+1)} = \left(1 - z_i^{(l)}\right) \odot H_i^{(l)} + z_i^{(l)} \odot \widetilde{H}_i^{(l+1)}$$

: weighted summation with update rate $z_i^{(l)}$ of temporary state $\widetilde{H}_i^{(l+1)}$ and previous state $H_i^{(l)}$

$$\widetilde{H}_i^{(l+1)} = \tanh\left(W \cdot \left[r^{(l+1)} \odot H_i^{(l)}, m_i^{(l+1)}\right]\right)$$

: temporary state is updated with product of forget rate $r^{(l+1)}$ and previous state $H_i^{(l)}$, and message state $m_i^{(l+1)}$

In this case, $x_t = m_i^{(l+1)}$

$$m_i^{(l+1)} = \sum_j H_j^{(l)}$$

$$z_i^{(l+1)} = \sigma\left(W_z \cdot \left[H_i^{(l+1)}, m_i^{(l+1)}\right]\right)$$: update rate

$$r_i^{(l+1)} = \sigma\left(W_r \cdot \left[H_i^{(l+1)}, m_i^{(l+1)}\right]\right)$$: forget rate

Codes available with scripts at https://github.com/SeongokRyu/augmented-GCN

Accuracy of different models

Accuracy in logP predictions

	FP - MLP	SMILES – CNN	Graph – GCN	SMILES – RNN1	SMILES – RNN2	SMILES – RNN3	Graph – GGNN	Graph - my best*
MAE	0.31	0.15	0.088	0.13	0.05	0.072	0.02	0.006
Std.dev	0.42	0.20	0.137	0.18	0.08	0.11	-	-

* : using modified self-attention, which used in the Transformer (SOTA at NLP), and gated skip-connection

Assignment #7

Implementation of GGNN

- In this class, TA overviewed possible statistical modelings for logP prediction
- In this week, we learned gated graph neural network (GGNN).
- Therefore, **implement the GGNN.**
- **Report your results - MAE, std. dev, and truth-prediction plot.**
- **In addition, please submit the report. Students have to describe the "meaning of statistical modeling, inputs for molecular applications, model architectures have been used and summarized results".**