Electronic Supplementary Material (ESI) for RSC Advances.
This journal is © The Royal Society of Chemistry 2022

Supplementary Information

Generation of novel Diels-Alder reactions using a generative adversarial network

Sheng Li, ‡[ab] Xinqiao Wang, ‡[b] Yejian Wu, [b] Hongliang Duan [bc] and Lan Tang *[a]

[a] *College of Pharmaceutical Sciences, Zhejiang University of Technology, Hangzhou 310014, P. R. China. E-mail: tanglan@zjut.edu.cn*

[b] *Artificial Intelligence Aided Drug Discovery Institute, College of Pharmaceutical Sciences, Zhejiang University of Technology, Hangzhou 310014, P. R. China.*

[c] *State Key Laboratory of Drug Research, Shanghai Institute of Materia Medica (SIMM), Chinese Academy of Sciences, Shanghai 201203, China.*

Sheng Li, XinQiao Wang contributed equally to this work.

Table of content

Section S1 Exploration of GAN model's hyperparameters

S1.1 Batch size

Batch size is the number of training samples employed by one GPU in one training step. In GAN model, batch_size is prescribed for the number of tokens (subwords) in one batch. We presented experiments with batch size 512.

Training throughput, the quantity of training data consumed by the training. It equals to the batch size multiplied by the computation speed. The model allows for using a high batch size on the premise that GPU cannot run out of memory. However, when the batch size exceeds a certain size, the performance of the model with bigger batch size does not increase markedly, as a result of mildly higher throughput. Consequently, it is good to know how a large batch size meets with your GPU.

S1.2 Drop out

The trained model prone to over-fitting when the parameters are too many or training samples are too few, which is a common phenomenon in machine learning. In order to solve the problem of overfitting, model integration is generally adopted, that is, multiple training models are combined. At this point, the time spent on training the model becomes a big problem. Not only does it take time to train multiple models, but it also takes time to test multiple models. Drop out can effectively alleviate the occurrence of overfitting and achieve regularization to a certain extent.

Drop out means that the neuron is removed with a probability of p during forwarding conduction. The network structure changes by randomly deleting a certain number of hidden neurons. The whole drop out process is equivalent to averaging many different neural networks. However, different networks produce different overfittings, and some "inverse" fitting cancels mutually to reduce overfitting on the whole. We presented experiments with dropout 0.1.

S1.3 Learning rate

The learning rate is a hyperparameter that guides how we should adjust the weights of the network through the gradient of the loss function. The lower the learning rate, the slower the loss function changes. While using a low learning rate ensures that we don't miss any local minima, it also means that it will take us longer to converge.

We optimized the network parameters with the Adam optimizer using a weight decay of 0.001 and a base learning rate of 0.01. Throughout the course of training, the learning rate was reduced by multiplying the base learning rate by 0.9 for each epoch according to the exponential learning rate schedule.