Challenges in Computer Aided Drug Design with Generative Adversarial Networks

İlayda Beyreli Kokundu
Department of Computer Engineering
Bilkent University
Ankara, Turkey 06800
ilaydabeyreli@gmail.com

***Abstract*—Drug design can be defined as creating new organic molecules that interact with biomolecules such as proteins in certain ways to affect their function in the biological system. It mainly involves designing molecules that possess specific physical and chemical properties. In the case of computer-aided drug design, we may illustrate molecules as graphs where the atoms are nodes, and the chemical bonds among them are edges. With such an analogy, deep generative networks (DGNs) show the potential of being powerful tools for drug and molecule design. Although one needs to use several heuristics to convert the traditional models that generate grid-structured data, such as images, recent research focuses more on drug design with generative models. This project aims to apply an approach to implement a generative adversarial network to predict the best planar molecule structure where the number of atoms per element and some desired properties are given as the input. This report contains information about the dataset and methods and discusses the challenges in developing, implementing, and training generative adversarial networks for computer-aided molecule and drug design. All scripts containing methods and models developed for this project can be is released at http://github.com/ibeyreli/molgen**

***Index Terms*—deep learning, drug design, generative adversarial networks**

I. Introduction

Computational approaches in drug design can help scientists with atomic level structure-activity relationship (SAR) used to facilitate the drug design process and minimize time and costs. [1]. With its advantages, computer-aided drug design (CADD) can improve drug resistance and design novel drugs for both known and new targets when combined with wet-lab techniques. There are two general CADD approaches: structure-based drug design (SBDD) and ligand-based drug design (LBDD). SBDD methods analyze the 3-dimensional structures of macromolecules to identify key sites and interactions, whereas LBDD methods focus on physiochemical properties. There are more than 30 non-commercial software CADD tools to model and analyze molecules, targets, and their relationship [2].

With the help of developments in machine learning, deep learning models are also employed for several steps in the drug design process and the classical tools. As research efforts have generated enormous amounts of data for drugs and drug candidates, machine learning methods have proven helpful in CADD to generate data-driven models [3]. Several models are developed to predict physicochemical properties, biological activities, such as ligand-binding activities, drug efficacy, and adverse effects. In addition to these discriminative models, several studies also utilized generative methods. In this project, the challenges in designing generative models for drug design are analyzed and discussed through a generative adversarial-based model called MolGen. MolGen models molecules as graphs, where the nodes are atoms and the edges are bonds among them. Hence, it predicts the best planar structure for a set of atoms by learning to construct a stable molecule by processing the adjacency matrices of previous molecules. In the following sections, several earlier studies in CADD are explained (Section II), the formal description of MolGen and the experimental setup is laid out (Section III), and the results are discussed (Section IV).

II. Related Work

Machine learning methods have been used in drug design for Quantitative Structure-Activity Relationship (QSAR) and ligand analysis. However, these methods were mainly based on classical ML methods such as random forests, and SVMs [3]. Recently, more computationally powerful architectures have been considered for drug design applications. For example, Jimenez-Carretero et al. [4] trained a convolutional neural network (CNN) to predict toxicity from images of DAPI-stained cells pretreated with a group of drugs with different toxic mechanisms. AtomNet [5] is another CNN-based model developed by Wallack et al. to predict the bioactivity of small molecules. This model takes 3D grids placed over ° co-complexes of the target proteins and small molecules sampled within the target's binding site. Then, after pre-processing, the input is processed with 3D convolutional layers.

Sequential models are also used for CADD. Using sequential representation of molecules in SMILES [6] notation, Gupta et al. [7] trained a RNN for *de novo* drug design. They aimed to enable virtual compound design without secondary or external activity prediction. Xu et al. [8] also developed a conditional RNN (cRNN) model to integrate the 3D structural information of the protein binding pocket for targeted molecule generation. However, models similar to these learn rules such as the SMILES grammar and atom ordering which are irrelevant to molecular structures. To overcome this bias, Yi et al. [9] proposed a method that combines graphs processing techniques, essentially graph convolutional networks GCNs, with RNNs. Although Yi et al.'s graph-based model outperforms SMILES-based models, there are still areas where it falls short. For example, RNNs cannot model long-term dependencies, which means they result in poorer for bigger molecules that consist of a high number of atoms. Both LSTMS and RNNs also are sensitive to different random weight initializations and easy to overfit. Finally, embeddings obtained from these models are not designed to be disentangled.

III. Methods

A. Dataset

ChEMBL is a database for small bioactive molecules curated from scientific literature [10]. It contains 2D structures and properties of more than 2 million drug-like molecules. The aim is to use the properties of these compounds to generate their 2D structures correctly. The closed formula of the molecules and their SMILES notations are obtained from this dataset. After finding the highest number of atoms from each element using closed formulas, fixed sized adjacency matrices, As, are generated in such a way that if element i and element j bonds in molecule m, then ${}^{m}A_{ij} = {}^{m}A_{ji} = 1$ and else ${}^{m}A_{ij} = {}^{m}A_{ji} = 0$ where ${}^{m}A$ is the adjacency matrix for molecule m. The bond information to generate such matrices is obtained from the SMILES notation. The redundant elements for each molecule are left disconnected in the adjacency matrix.

For the feature vector x, the number of atoms for five core elements of hydrocarbons, carbon (C), oxygen (O), hydrogen (H), nitrogen (N), and fluoride (F), are obtained. For molecule m, ${}^{m}x$ contains the number of C, O, H, N, and F atoms in the molecule. While this is not the most practical setup, it is believed to be the most basic feature vector to be fed to the network in the initial phase to analyze how the generative network works and what it learns.

B. MolGen

MolGen is a multilayer perceptron (MLP) based generative adversarial network which takes desired features for the molecule and the number of atoms for each element and generates an adjacency matrix representing the planar structure of the molecule. The model consists of 3 modules shown in Figure 1: the module that generates conditional code, the generator, and the discriminator. In the simplest case, the conditional code module (CCM) applies identity transform. However, it is also possible to learn a particular embedding. The generator and the discriminator are three-layer MLP-based modules, where batch normalization and Leaky RelU activation function with a negative sloped of $m = 0.2$ is applied after each linear hidden layer. Also, dropout with probability $p = 0.4$ is used for the discriminator.

MolGen uses mean square error for the discriminator loss. The generative loss consists of three terms: the adversarial loss L_{adv}, the symmetry loss L_{sym}, and the bond loss L_{bond}. The adversarial loss is similar to the discriminator loss, where the mean square error between the output of the discriminator for the subject image and 1.0, which is the label for real molecules. The symmetry and bond losses incorporate the domain knowledge into the training process. L_{sym} ensures the generated matrices are symmetric, i.e., the generated molecule graphs are un-directed. To achieve this, the generated matrix, ${}^{g}A$ is subtracted from its transpose as in Eqn. 1. If the adjacency matrix of m is symmetric, then ${}^{m}A_{ij} = {}^{m}A^{T}_{ij}$ for all i and j, and $L_{sym}(m) = 0$.

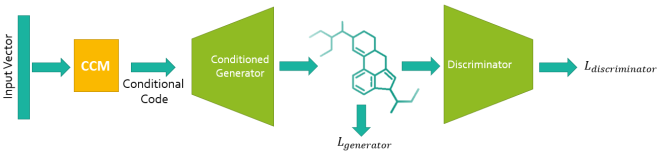

Figure 1. MolGen framework.

$$L_{sym}(m) = \sum_{i}\sum_{j} |{}^{m}A_{ij} - {}^{m}A^{T}_{ij}| \tag{1}$$

From chemistry, it is fundamentally known that each atom has a fixed number of valence electrons, i.e., the maximum number of bonds it can make is predefined. For example, a carbon atom in the molecule should not be allowed to have more than four bonds with different atoms, as it has four electrons in its valence band. Therefore, a reference vector, v, that represents the number of bonds each atom can make is used such that $v_i = 1/(\text{total number of bonds atom } i \text{ can make})$. Note that all adjacency matrices in the dataset are prepossessed so that the same row corresponds to the same element for all, and v does not change. Hence, the formulation for L_{bond} is given in Eqn. 2 where $\bar{y}$ represents the row sum of ${}^{m}A$, as in Eqn. 3. The final form of the generator loss is given in Eqn. 4.

$$L_{bond}(m) = |\bar{y}v| \tag{2}$$

$$\bar{y}_i = \sum_{j} A_{ij} \; for \; j \neq i \tag{3}$$

$$L_{gen} = L_{adv} + L_{sym} + L_{bond} \tag{4}$$

C. Learning and the Validation Setting

IV. Results & Discussion

Training and validation losses are tracked to analyze the training process. The learning curves for the most recent session are presented in Figure 2. It is observed that the discriminator quickly overfits with losses around 1.7×10^{-6}. However, the generator takes longer to converge. Although the training loss is around 0.99, the validation loss starts from 1000.0 and goes down to 35. When the generator losses are analyzed separately during training, it is seen that the symmetry loss has the lowest contribution with less than 1%. This is due to the fact that the symmetry is relatively easy to be captured by the generator. Hence, the model learns it earlier, which quickly decreases the relative loss very low.

Both the bond loss and the adversarial loss in training are around 10 to 20. Compared with the discriminator loss, these

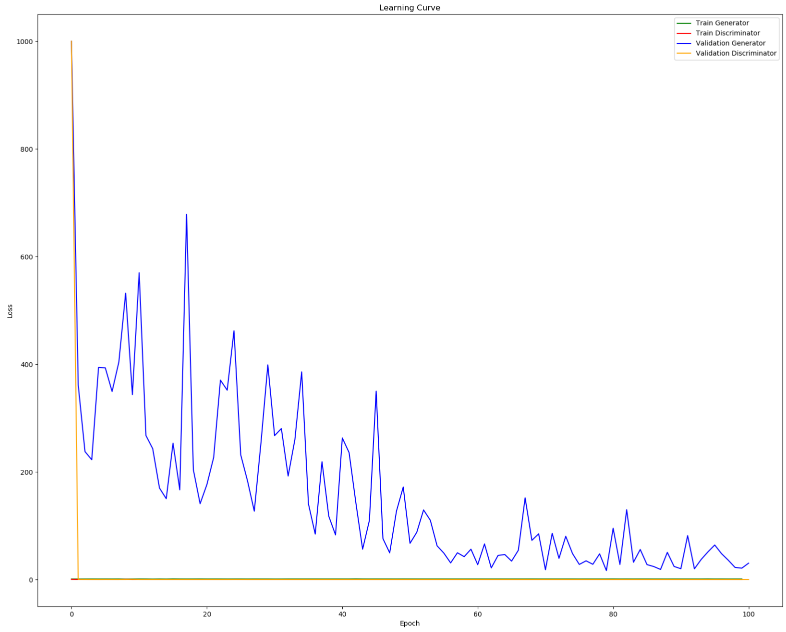

Figure 2. Learning curves. Green: Training loss for the generator, Blue: Validation loss for the generator, Red: Training loss for the discriminator, Yellow: alidation loss for the discriminator.

losses are significantly high, which probably causes disparity among the modules. Hence, the first challenge with the model is that the generator cannot catch up with the discriminator, which is a general pitfall for adversarial networks. However, this difficulty can be fixed with better parameter optimization and an enlarged training set size.

The second challenge with deep molecule generation with adversarial networks is due to the excessive space complexity. Theoretically, the most extensive molecule in this dataset can contain 6000 atoms, which means each adjacent matrix is set to be $6000x6000$ leading to 36MB per matrix. Compared to a $32x32$ RGB image of size 24kB, the graphs require excessive memory to be stored and processed. One should also note that the feature vector is kept as simple as possible in this work. No realistic property for molecules such as the oiling point or targeting a specific protein is passed to the model up to this point. It means that including these properties will increase the model size, as the problem will be more complex, and the memory requirement will be increased more. As bigger GPUs can always be used, it is important to consider the batch size in those experimental setups. As the hardware reaches its limits, one may need to use synchronized batch normalization [11]. Even then, the small-batch size can cause unstable learning.

Another candidate solution could be using a CNN-based generator and discriminator, as they require fewer parameters than MLPs to optimize for problems of similar complexities. However, the adjacency matrices are very sparse. Also, there is no correlation between the place of a node in the adjacency matrix and its neighbors. Therefore, convolutional layers' performance would not meet the expectations if they were used. One might be able to use this sparsity of the networks to decrease memory requirements. For example, only the lower triangular region of the adjacency matrices can be considered. This approach not only reduces the sample size but it decreases the number of neurons in each layer, enabling stacking more layers to solve more complicated tasks. However, even with the lower triangular region process only, the input would still be sparse, meaning that the majority of the neurons would be suppressed, and training all neurons would require a long time.

In addition to the challenges in balanced learning and memory requirements, interpreting the evaluating the performance of molecule generation models is also problematic. Firstly, it is unknown how well the specially designed loss functions represent the desired outcome. Hence, there is no systematic way to ensure that model learns what it is designed to learn. There is also no guaranteed computational way to measure how realistic the generated molecules are. Even with carefully carried out biochemical and clinic experiments, the evaluation process is time-consuming and expensive.

V. Conclusion

Computer-aided drug design is an open research area, and with the recent improvements in machine learning, researchers use more and more complex models to improve performance and decrease costs in drug design. However, even with today's sophisticated methods, there are still challenges that need to be mastered. Unstable learning due to intrinsic characteristics of generative models and data sparsity, excessive memory requirements, and lack of proven validation methods are the three of those challenges that can be identified within the scope of the project. Hence, the future work of this project is to study more and propose more delicate computational solutions to achieve better results in CADD.

References

[1] W. Yu and A. D. MacKerell, "Computer-aided drug design methods," in *Antibiotics*, pp. 85–106, Springer, 2017.

[2] S. Suat, "Molecular modelling and computer aided drug design: The skill set every scientist in drug research needs and can easily get," *Hacettepe University Journal of the Faculty of Pharmacy*, vol. 40, no. 1, pp. 34–47, 2020.

[3] L. Zhao, H. L. Ciallella, L. M. Aleksunes, and H. Zhu, "Advancing computer-aided drug discovery (cadd) by big data and data-driven machine learning modeling," *Drug discovery today*, 2020.

[4] D. Jimenez-Carretero, V. Abrishami, L. Fernandez-de Manuel, I. Palacios, A. Quílez-Álvarez, A. Díez-Sánchez, M. A. Del Pozo, and M. C. Montoya, "Tox_ (r) cnn: Deep learning-based nuclei profiling tool for drug toxicity screening," *PLoS computational biology*, vol. 14, no. 11, p. e1006238, 2018.

[5] I. Wallach, M. Dzamba, and A. Heifets, "Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery," *arXiv preprint arXiv:1510.02855*, 2015.

[6] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[7] A. Gupta, A. T. Müller, B. J. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider, "Generative recurrent networks for de novo drug design," *Molecular informatics*, vol. 37, no. 1-2, p. 1700111, 2018.

[8] M. Xu, T. Ran, and H. Chen, "De novo molecule design through the molecular generative model conditioned by 3d information of protein binding sites," *Journal of Chemical Information and Modeling*, vol. 61, no. 7, pp. 3240–3254, 2021.

[9] Y. Li, L. Zhang, and Z. Liu, "Multi-objective de novo drug design with conditional graph generative model," *Journal of cheminformatics*, vol. 10, no. 1, pp. 1–24, 2018.

[10] A. Gaulton, A. Hersey, M. Nowotka, A. P. Bento, J. Chambers, D. Mendez, P. Mutowo, F. Atkinson, L. J. Bellis, E. Cibrián-Uhalte, *et al.*, "The chembl database in 2017," *Nucleic acids research*, vol. 45, no. D1, pp. D945–D954, 2017.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.