ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7. Available from http://www.i6doc.com/en/.

Improvement on Generative Adversarial Network for Targeted Drug Design

Beatriz P. Santos, Maryam Abbasi, Tiago Pereira
Bernardete Ribeiro and Joel P. Arrais *

Department of Informatics Engineering (DEI)
Center for Informatics and Systems of the University of Coimbra (CISUC)
University of Coimbra, Coimbra, Portugal

Abstract. This paper provides a generative network framework that can replicate the molecular space distribution to satisfy a set of desirable features. The approach incorporates two effective machine learning techniques: an Encoder-Decoder architecture that converts the string notations of molecules into latent space and a generative adversarial network to learn the data distribution and generate new compounds. We train this joint model on a dataset that includes stereo-chemical information. The results show an improvement in the Encoder-Decoder performance, reaching 89% of correctly reconstructed molecules. The framework can generate a wide variety of compounds biased towards specific molecular properties using Transfer Learning.

1 Introduction

Drug discovery is a highly time-consuming, complex, and expensive process with low rates of success. Recent estimates point that to develop a new drug, an average $1.8 billion and twelve years are required [1]. This is partly due to the high dimensionaly of the chemical space, which has been estimated to include between 10^{33} and 10^{60} synthetically accessible molecules [2]. Evaluating the entire chemical space is prohibitively expensive, so it is of the utmost importance to find strategies to narrow down the search space. Deep Learning (DL) techniques, mainly generative models, have emerged as a promising solution for *de novo* drug design, whose aim is to generate novel molecules with specific predefined properties [3]. This process can typically be seen as two-fold: first, creating a general model that learns the training data distribution and then optimizing this model to the desired property, usually with either Reinforcement Learning (RL) or Transfer Learning (TL).

Earlier approaches to solving this problem resorted to the use of Recurrent Neural Networks (RNNs), which can learn the syntax of sequences of data. They were used to generate new molecules represented as Simplified Molecular Input Line Entry Strings (SMILES), which would then be optimized with RL [4] or TL[5]. However, this approach can suffer from exposure bias [6] which prompted the appearance of other DL-based alternatives for the targeted generation of compounds. Gomez-Bombarelli et al. proposed the use of a Variational Autoencoder (VAE) that transforms the discrete data into a latent real-valued

*This work has been supported by the Portuguese Research Agency FCT, through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER-029266).

continuous vector [7]. The VAE is trained together with a Predictor that, given a latent vector, predicts specific properties. In a similar manner, Blaschke et al. compared VAEs with adversarial autoencoders and showed that, in both cases, the latent space preserves the chemical similarity of the input molecules [8]. Prykhodko et al.used an autoencoder to find a numerical representation of the SMILES Strings that would then be used to train a Generative Adversarial (GAN) Network[9].

The previous work does not take into account stereochemistry, which is a significant factor in drug design, nor does it consider the diversity of the produced molecules. In this work, we perform an exhaustive grid search of the Encoder-Decoder model resulting in the use of an embedding layer for the encoder. We include in its training data SMILES strings that contain stereo-chemical information, which is vital when working in drug design but often overlooked due to its higher complexity. A GAN is then trained on the latent space vectors created by the Encoder-Decoder model, and its capability to create diverse molecules is evaluated. We also show that the framework can be optimized towards specific properties using TL.

2 Methods

The general framework of this study is illustrated in Figure 1 and composed of an autoencoder, more specifically, an encoder-decoder architecture based on RNNs [10, 11], and a Wasserstein GAN with gradient penalty [12]. The encoder-decoder architecture allows the model to learn a context vector (Fig.1-C) that summarizes the SMILES strings in such a way that it can be reconstructed only from this context vector. By passing a dataset that consists of SMILES through the encoder (Fig.1-A), an equivalent dataset made up of context vectors is obtained. This new dataset is used as real data to train the GAN so that, once trained, its generator is able to generate new samples from the same distribution as the context vector's dataset. These samples are then passed through the decoder (Fig.1-B) in order to obtain the corresponding SMILES strings. By combining these two models, it becomes possible to train the GAN, surpassing the differentiation problem associated with categorical data, such as SMILES strings, that would arise [13].

2.1 Encoder-Decoder Model

The encoder-decoder model is an autoencoder that works with sequences of data by resorting to recurrent networks. The encoder-decoder architecture allows the learning of a context vector. The model summarizes the input SMILES in such a way that can reconstruct them only from the context vector. The encoder, Fig. 1-A, contains an embedding layer and two bidirectional LSTM layers with batch normalization in between. From these bidirectional layers, the final cell and hidden states, from both directions, are retrieved and then concatenated. Then the result is passed through a dense layer with a size equal to the desired length of the context vector. Next, there is a dense layer followed by batch normalization and a Gaussian noise layer (during training) to make the model more flexible and the context vector more robust. For the decoder, Fig 1-B, the context vector serves as input to four independent dense layers whose goal

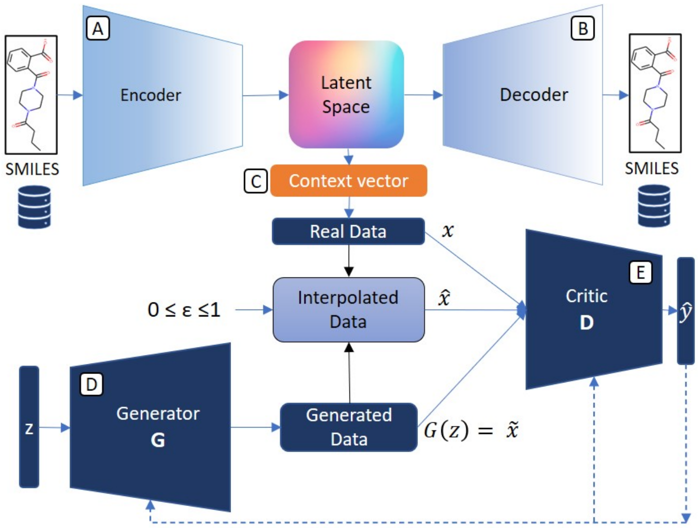

Fig. 1: The general workflow of the proposed model that composed of an Encoder-Decoder to convert SMILES to latent space in-order to be used as real data in the training of a WGAN-GP network that comprises a Generator and Critic.

is to reconstruct the hidden and cell states that will be given as initial states to two LSTM layers. These stacked LSTM layers, with batch normalization in between, are followed by a dense layer with *softmax* activation function so that it outputs the probabilities associated with the next token. It should be noted that, during training, the teacher's forcing algorithm is executed, and the target output of the decoder is the same as its input but shifted by one time step. The complete model is trained using the categorical cross-entropy between the input and the predicted output as the loss function. The network's weights are then updated by the Adam optimizer, considering the gradient of the loss. Note that the decoder employs the embedding layer to convert the input data, while a one-hot encoding layer is used for the decoder due to its improved performance.

2.2 Generative Adversarial Network

The second part of the framework is based on a Wasserstein GAN with gradient-penalty (WGAN-GP). This model has better performance and stability during the training process [12] compared to the traditional GAN as proposed in [13]. The WGAN-GP contains two neural networks with opposing objectives: a generator (Fig.1-D) and a critic (Fig.1-E). The critic's goal is to distinguish between generated $\tilde{x}$ and real data x while the generator aims to fool the critic into believing that its samples are real. Since the output of the critic is not bounded, to prevent its gradient from deviating from 1, a gradient penalty (weighted by λ) is included in the loss function (L). The following equation shows the loss function where the third term corresponds to the gradient penalty:

$$L = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\left[D(G(\boldsymbol{z}))\right] - \mathbb{E}_{\boldsymbol{x}\sim p_{data}}\left[D(\boldsymbol{x})\right] + \lambda \mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]$$

$$\text{If} \quad \tilde{\boldsymbol{x}} = G(\boldsymbol{z}) \quad \text{then} \quad \hat{\boldsymbol{x}} = \epsilon * \tilde{\boldsymbol{x}} + (1-\epsilon) * \boldsymbol{x} \quad \text{with } 0 \leq \epsilon \leq 1$$

where ϵ is uniformly sampled between 0 and 1. In practice, the gradient penalty amounts to creating a set of interpolated samples $\hat{\boldsymbol{x}}$ (each using a real sample $\boldsymbol{x}$, a generated sample $\tilde{\boldsymbol{x}}$ and ϵ) that result from randomly choosing points that lie on the lines that connect the batch of real samples to the batch of fake samples and evaluating its gradients. The gradient penalty loss term returns the squared distance between the gradient calculated at the interpolated points and 1, and so, it penalizes the critic whenever its gradient deviates from 1, therefore enforcing the 1-Lipschitz constraint [12].

3 Experimental Analysis and Results

3.1 Dataset and Preprocessing

The dataset used throughout the experimental analysis includes 500,000 SMILES strings of drug-like small molecules retrieved from the ChEMBL database [14] and from the dataset used by [5] which contains biogenic structures. All the SMILES strings are canonicalized, and there are no duplicates. The SMILES strings are preprocessed by being tokenized character by character, adding ‘G’ as the first token of each SMILES and ‘A’ at the end and for padding. The SMILES are then either One-hot Encoded (OHE), where each token becomes a binary vector or passed through an embedding layer that converts each token into a dense vector that is learned by the model.

3.2 Validation Strategy

The encoder-decoder is a deterministic model; therefore, it is evaluated based on the percentage of molecules that can correctly reconstruct. The generated SMILES are syntactically and biochemically validated by RDkit [15]. From the group of valid molecules, its uniqueness is also computed as the percentage of different generated valid molecules. Since it is not just important that the generated molecules are unique but also that they are diverse, the *Internal Diversity* (diversity between the set of generated molecules) and *External Diversity* (diversity between the set of generated molecules and the set of training molecules) are also computed using the Tanimoto Similarity, T_s, that computes the similarity between two molecules represented by their circular fingerprint; $1 - T_s$ is the Tanimoto Distance, and hence the diversity is defined as the average of the tanimoto distance between every pair of molecules in the datasets.

3.3 Experimental Analysis on Encoder-Decoder

Several architectures for the Encoder-Decoder were evaluated, but the one that yielded the best results includes an encoder that contains two bidirectional LSTM layers with 512 units each, 256 for each direction. Both the embedding dimension and latent dimension were set to 256. All the batch normalization layers had a batch normalization momentum of 0.9. The Gaussian noise layer added noise with a standard deviation of 0.1. The model was trained using the Adam optimizer with a learning rate of 0.01, a batch size of 128, and the total number of epochs was set to 100, but only the best models regarding the validation loss were kept (10% of the training data was set as validation data). Table 1 summarizes the results for training dataset with 100,000 and with 500,000 SMILES Strings. In both cases, the model reaches high percentages of correctly reconstructed molecules for the train and test sets. The model trained with 500,000 molecules learning to generalize better (with 89.2% and 89.0% for the train and test set, respectively). Molecules that are correctly reconstructed are automatically valid. Validity (evaluated by RDkit [15]) is constantly higher than the percentage of correct reconstruction, which means that some molecules are reconstructed into valid molecules but not the intended ones.

#Training Data	%Correct R. (train set)	%Correct R. (test set)	%Validity (train set)	%Validity (test set)
100,000	88.6	86.8	89.7	89.1
500,000	**89.2**	**89.0**	**89.9**	**89.8**

Table 1: Performance of the Encoder-Decoder model for 100,000 and 500,000 training dataset.

3.4 Experimental Analysis on WGAN-GP

The implemented WGAN-GP comprises a critic made up of three dense layers with 256 units and with Leaky-Relu as the activation function (with $\alpha = 0.3$), except for the final layer, which does not include an activation function. A vector of 64 dimensions is drawn from the uniform distribution and then passes through the Generator that contains five dense layers with 256 dimensions each, except for the first layer, which contains 128 dimensions; between these layers, Leaky-Relu activation function ($\alpha = 0.3$) and batch normalization (momentum of 0.9) layers are applied. Both the Generator and the Critic are trained using the Adam optimizer with a learning rate of 0.0001 and include drop-out layers with a value of 0.2. The WGAN-GP was trained on 100,000 molecules and for 10,000 epochs. The results of sampling 1,000 valid molecules using the full framework are presented in table 2. As proof of concept and to evaluate the ability of the model to mimic the desired distribution, the model was also biased through TL using a dataset with SMILES strings and their corresponding affinities for the Kappa opioid receptor (KOR) (CHEMBL identifier 237) as measured by the pIC50. This dataset contained only SMILES with a pIC50 of less than 7. The biased model was trained for an additional 4,000 epochs with this dataset, 1,000 valid molecules were sampled, and their pIC50 were predicted. As observed in Table 2, the biased model retains high levels of uniqueness and diversity. Fig.2 shows that this model effectively shifts the distribution to encompass more uniformly the values in the range of 5 to 7.

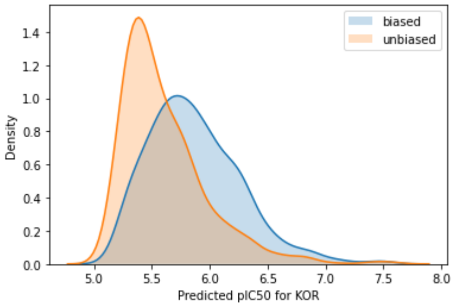

Fig. 2: Predicted pIC50 for 1,000 molecules sampled from the biased model.

4 Conclusions

In this work, we showed improvements in an Encoder-Decoder architecture that, joined with a WGAN-GP, can be trained to generate new molecules that

Model	%Uniqueness	Internal Diversity	External Diversity
unbiased	100.0	0.8907	0.8914
biased	99.3	0.8777	0.8764

Table 2: Performance of the WGAN-GP model for the unbiased and biased model.

include stereo-chemical information with high diversity (0.89) and uniqueness (100.0%). Moreover, this framework can also be optimized to generate molecules with bespoken properties using Transfer Learning while keeping high values of both diversity and uniqueness, as we've shown by biasing the general model to generate molecules with a pIC50 of less than 7 for the KOR receptor.

References

[1] S. Sinha and D. Vohora. Drug discovery and development: An overview. *Pharmaceutical Medicine and Translational Clinical Research*, pages 19–32, 2018.

[2] P. G. Polishchuk, T. I. Madzhidov, and A. Varnek. Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, 27 (8):675–679, 2013.

[3] G. Schneider and U. Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005.

[4] T. Pereira, M. Abbasi, B. Ribeiro, and J. P. Arrais. End-to-end deep reinforcement learning for targeted drug generation. In *In ICCBB 2020: 2020 4th International Conference on Computational Biology and Bioinformatics (ICCBB 2020)*, 2020. doi: 10.1145/3449258.3449260.

[5] S. Zheng, X. Yan, Q. Gu, Y. Yang, Y. Du, Y. Lu, and J. Xu. Qbmg: quasi-biogenic molecule generator with deep recurrent neural network. *Journal of cheminformatics*, 11 (1):1–12, 2019.

[6] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

[7] R. Gómez-Bombarelli, J. N Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, Dennis Sheberla, Jorge A., T. D Hirzel, R. P Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[8] T. Blaschke, M. Olivecrona, O. Engkvist, J. Bajorath, and H. Chen. Application of generative autoencoder in de novo molecular design. *Molecular informatics*, 37(1-2): 1700123, 2018.

[9] O. Prykhodko, S. V. Johansson, P. Kotsias, J. Arús-Pous, E. J.k Bjerrum, O. Engkvist, and H. Chen. A de novo molecular generation method using latent vector based generative adversarial network. *Journal of Cheminformatics*, 11(1):1–13, 2019.

[10] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[11] I. Sutskever, O. Vinyals, and Q.V. Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[14] A. Gaulton, L.J. Bellis, A.P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[15] G. Landrum et al. Rdkit: Open-source cheminformatics, 2006. https://www.rdkit.org/.