Using invertible plugins in autoencoders for fast and customizable post-training optimization

Domonkos Pogany
Budapest University of Technology and Economics
Department of Measurement and Information Systems
Budapest, Hungary
pogany.domonkos@gmail.com

Peter Sarkozy
Budapest University of Technology and Economics
Department of Measurement and Information Systems
Budapest, Hungary
psarkozy@mit.bme.hu

***Abstract*—One of the main motivations for modern drug research is the production of new compounds that act as drugs, however developing a new drug is an excessively time and resource intensive process. Deep generative neural networks might provide a solution. With their help, we may be able to search in a continuous latent space to find drug molecules that are not yet known but have suitable chemical and structural properties (e.g. solubility, interaction with a given target protein).**

In this paper, we propose a model which can generate novel drug candidates, that are suitable for a pre-specified objective function of arbitrary properties. The model consists of a generative network and a predictor. The former is an autoencoder which utilizes attention to handle the textual representation of molecules, while the latter uses matrix factorization to predict drug-target interactions (DTI). With a genetic algorithm we can generate novel compounds from the continuous latent space, but if there are changes in the objective function, we may need to train the whole model again. This problem is typical of conditional generative models, to address it, we separated the predictor from the pretrained autoencoder thus forming the plugin. In addition to getting a flexible architecture without any deterioration in the so far achieved results, our model can also be used in a distributed setup by concatenating the plugins. In this way, the objective function can be broken down to smaller subtasks, which can be solved by different plugins without sharing any data.

***Index Terms*—molecule generation, autoencoder, transformer, genetic algorithm, plugin, DTI**

I. Introduction

Finding novel drugs is a difficult challenge, despite all the efforts there are only 10^8 molecules that have been synthesized so far [1] out of the estimated possible 10^{60} drug-like compounds [2]. Developing a new drug is at least a decade-long task usually consisting of three main steps. The first step is the selection of the target proteins, then comes the selection and optimization of the drug compounds, and the last phase is the testing of the drugs. Generative artificial intelligence models may help in reducing the cost and time required for the selection and optimization of drug candidates.

Generating novel drug candidates is an active research field. Most existing methods work with a textual representation of molecules such as the Simplified molecular-input line-entry system (SMILES), which represent molecules as a string. Methods based on a variational autoencoders (VAE) [3] are very common. They can be used to convert the discrete space of the textual representation into a continuous latent space in which the search and optimization tasks can be accomplished more easily. Instead of the textual representation, working on the molecular graphs is also possible [4], this guarantees that the generated molecules will always be syntactically valid. Other methods construct novel and valid molecules step by step with the use of reinforcement learning [5], or by starting from a known compound and optimizing it according to a molecule property using a generative adversarial network (GAN) [6]. To handle the input representation, most models use recurrent layers. Using attention layers instead, to speed up the training process is also becoming a popular choice. Generating compounds with transformers [7] have shown promising results lately [8].

The methods mentioned so far can all generate valid molecules according to some chemical properties, however, they do not in themselves provide a solution for generating a pharmacokinetically active molecule that interacts with the target protein. Measuring all interactions between every compound-target pair is extremely costly, thus we need to use estimations. Neural networks can also be used for this task, for instance we can train a convolutional network to take a molecule and a target as an input and predict the interaction on the output [10]. By combining the generative and the predictive models above, in theory we are able to generate drug candidates for a given target.

Recent research places great emphasis on the development of a selective binding profile. The goal is to generate molecules that bind not only to one but to several target proteins. Moreover, there are targets we want to avoid binding to because of the possible side effects. Knowing the targets, we can now utilize the methods mentioned above to generate drug candidates. For instance, a VAE based method was presented to find compounds to treat psychiatric disorders by generating novel targets to the indicated genes [11].

Using these models in practice, however, introduces new problems. When modifying the objective function, such as introducing a new target, we might need to train the entire model again which could take weeks. Consequently, we need to utilize pre-trained models. More than that, most of the times, we do not have sufficient training data available. Usually compounds and even targets under research are kept highly

DOI 10.3311/MINISY2022-017

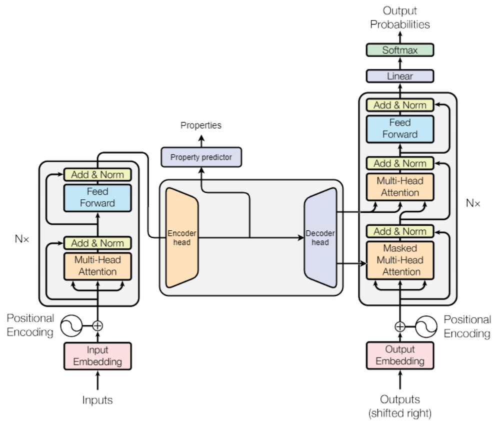

Fig. 1. Transformer based autoencoder [7].

confidential by pharma entities. The goal is to develop methods that allow partners to benefit from each other's knowledge without having to share their data and results.

In this paper we propose a model which can provide a solution to the above-mentioned problems. Besides being able to generate novel molecules for a multi-target objective function, training the model to a new, and previously not included target is highly efficient, and it can even be used in a distributed environment without sharing any valuable data.

II. Implemented Methods

A. Transformer-based autoencoder for novel compounds

We chose to work with the SMILES representation of molecules. We used a transformer network to handle the textual representation, and to be able to generate new compounds, we converted it into an adversarial autoencoder (AAE) [12] which uses a Gaussian prior.

First, the input molecules are encoded into a continuous latent space, to achieve this we put an extra attention layer in top of the original transformer blocks. From there chemical properties are predicted using fully connected layers. By propagating the error of the predictor back through the encoder, molecules with similar properties are mapped close together in the latent space. The decoder consists of transformer decoder blocks, each of them receives information from the latent space through fully connected layers. Two different representations are decoded from the latent space, one going into the cross attention layer and another into the self attention layer. The latter is concatenated into the decoder-side representations of the characters from the previous layer, thus forming a pseudo attention layer [13]. Fig. 1 shows the architecture of the model without the discriminator network.

From now, the problem of generating molecules to a given target function can be solved by searching in the latent space. To perform the search, we implement a genetic algorithm. The individuals are the molecules, their genes are their latent representations, and the fitness function is the objective function which we want to maximize. With this choice we can easily implement mutation as additive noise, and crossover as interpolation in the latent space.

B. Matrix factorization plugin for generating to a given target

To be able to generate drug candidates to a given set of proteins, first we need to somehow obtain and process DTI data. We chose to work with the ExCAPE-DB dataset [14] as it contains ∼600K molecules and ∼1300 targets. To utilize this extra information, we use an interaction predictor the same way as the property predictor. It solves the problem of the prediction with matrix factorization by taking the product of the latent molecule representation and the learned target embedding. As DTI data is often imbalanced with respect to active vs. inactive molecules, we weighted the binary cross-entropy loss with the imbalance ratio.

With the extensions discussed earlier we can generate molecules to maximize an objective function which consists of the predicted binding affinity to multiple target proteins. However, the end-to-end training of the model has a lot of drawbacks in practice. As already mentioned, it is inefficient when generating to a new target protein, and the sparsity of the available interaction data can also be a problem. To alleviate these problems, we pretrain our model. There are several pre-training techniques, and we have further investigated and developed a method created for conditional autoencoders [15]. The main idea is to pretrain the autoencoder and freeze all the weights, then attach a plugin into the bottleneck, which is a smaller autoencoder using the latent representation as input and has its own, inner latent space. This way the reconstruction and the conditional generation tasks can be separated, and when we receive a new condition, we only need to change the plugin. For the reconstruction task, we merged the GuacaMol, ExCAPE-DB and the DrugSpaceX [16] datasets, thereby resulting in 9.2 million training molecules for the model. After pretraining, the ExCAPE-DB dataset can be used to train the plugin and the interaction predictor. The predictor now uses the inner latent space as the input therefore the interaction data is encoded into it. Due to the small number of parameters in the plugin, we can learn a novel target in a relatively short time, moreover we can achieve good generalization even with a small amount of available interaction data. Plugins simplify post-training on any subset of the full training dataset. The modified architecture is shown in Fig. 2, the encoder and decoder of the plugin both consists of fully connected layers with a leaky ReLU between them.

After training the plugin the model was able to predict interactions from the inner latent space but lost the ability to reconstruct molecules from the pretrain dataset (from 95% to 10%). To solve this, we replaced the plugin decoder with the inverse of the plugin encoder. Therefore, the reconstruction power of the pretrained model is preserved. Moreover, training the plugin is greatly accelerated because it is no longer has trainable parameters in the decoder, so the decoder of the plugin and the transformer is not utilized during training. The

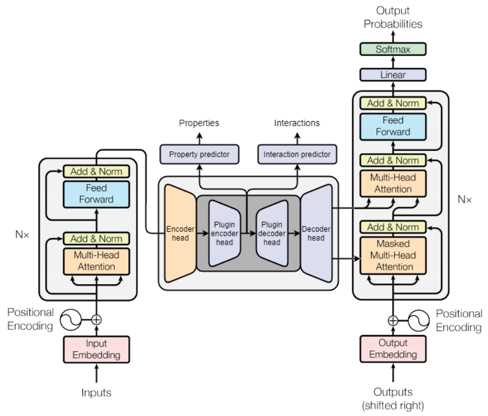

Fig. 2. Pretrained autoencoder extended with a plugin.

invertibility of the plugin carries a few restrictions[1], and the model becomes increasingly sensitive to noise. The hidden representation of the encoder in layer l, $x^{(l)}$ can be calculated as

$$x^{(l)} = g(W^{(l)}x^{(l-1)} + b^{(l)}) \quad \text{with} \quad g(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \frac{x}{2}, & \text{otherwise,} \end{cases} \tag{1}$$

where $b^{(l)}$ is the bias, $W^{(l)}$ is the weight matrix in layer l, and g denotes the activation function. Due to invertibility $W^{(l)}$ must be a square matrix, g must be invertible and its range must not be finite, hence we use a leaky ReLU. This way a layer in the decoder takes the form below

$$x^{(l)} = W^{(l)+}(g^{-1}(x^{(l-1)}) - b^{(l)}), \tag{2}$$

where $W^{(l)+}$ denotes the Moore–Penrose pseudoinverse of the weight matrix[2].

However, large weights in the matrix amplify sensitivity to noise in the plugin, even a small amount of noise added to the inner latent space resulted in high distances in the outer space. We examined the weight matrices and noticed that this can be a result of the high condition numbers ($> 1e4$). The sensitivity to noise can be handled by controlling the condition number of the components in the plugin network. Conventional regularization methods such as L1 and L2 loss were unsuccessful in keeping the condition number low, so we added the condition number directly to the loss function as a regularization parameter, and determined that condition numbers < 20 were still stable with respect to noise. The condition number of the weight matrix upon initialization was also controlled. We also found that using a steeper ReLU slope leads to a more stable training.

[1]The invertibility of the encoder was also the main reason behind choosing AAE, because the reparametrization trick used in a VAE is not invertible.

[2]It is very unlikely that the weight matrix is not invertible, and there are also methods with which we can assure the invertibility of the learned weights, for example learning the QR decomposition of the matrices.

The components of the loss function used for training the autoencoder plugin are the weighted and weight-normalized sum of the following:

- categorical cross-entropy of the reconstruction,
- binary cross-entropy of the DTI classification,
- MSE of the molecule property predictors,
- L2 weight regularization,
- condition number regularization: $max(0, C-20)^2$, where C is the condition number of the weight matrices,
- adversarial loss.

With these modifications the model was indeed able to generate hundreds of novel molecules from the inner latent space that were suitable for a given objective function.

C. Serial plugin for generating in a distributed environment

With this flexible architecture when a new protein is added to the targets only the plugin needs to be retrained. We can go even further by having several plugins to several targets. This way when a new target is required, a new plugin should be trained, the other plugins remain intact, so the time previously invested in them is not wasted. When generating we now have several plugins, each contributes to a couple of predicted properties or interactions. This is equivalent to connecting them serially, as inversion preserves the input. Fig. 3 shows the serial architecture.

To implement the method, besides connecting the plugins the optimization process based on the genetic algorithm needs to be modified. The fitness function can be constructed from the output of the plugins. The only difference with the one plugin case is that there are now more latent spaces, so we need to come up with a new crossover method. One solution is for each plugin to generate some individuals to form the new generation, but it can also work if we generate the next generation with a different plugin in each round. Selection and mutation can be performed as usual in the outer latent space.

The separability of the architecture and the versatility of the genetic algorithm allows the model to be used in a distributed environment. The obvious solution is to give every partner a plugin to work with, this way they only need to agree on the

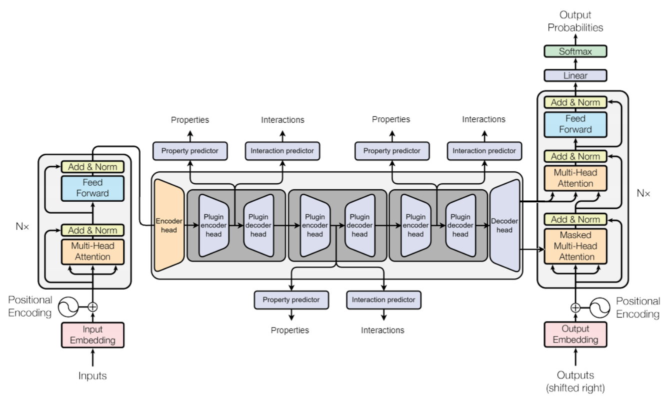

Fig. 3. Autoencoder with several plugins connected in series. Invertibility permits the usage of any number of plugins in serial connection.

pretraining data set. During generation they need to share their own calculated part of the fitness function and the molecules generated for the next generation. The sensitive training data, the known interactions, the parameters and architecture of the model, and even the methods using for prediction can remain hidden, mitigating the most common federated attacks [18]. More than that, the partners do not even need to share their selected target proteins if they are using a more abstract fitness function, e.g. avoiding a given side effect instead of not to bind to a given set of targets.

III. Results

We trained the model on the GuacaMol dataset [9] which contains roughly 1.6 million drug-like molecules, to evaluate the performance of the autoencoder model before attaching plugins. After the optimization of the hyperparameters, we managed to reach a remarkably high, 97.2% reconstruction on the test molecules and obtained similar benchmark results as the other models reported by the GuacaMol benchmark [9]. Using a sampling method of a random noise vector with $\mu = 0.0$, $\sigma = 0.25$ around test data, we achieved a validity of 96.11%, a uniqueness of 99.97%, a novelty of 97.18%, and the KL and FCD scores were 0.9688 and 0.8505 respectively. To test the plugins, we generated drug candidates to bind to specific targets. We separated the problem into two subtasks, to generate drug-like and synthesizable molecules and to generate molecules which can bind to a set of related protein targets, that often have strong cross-binding for most known drug compounds. First, we had pretrained the model on our merged dataset, after that, two plugins were trained. The one, which was trained on the GuacaMol dataset was responsible for the first task. To achieve this, we used a property predictor to predict the quantitative estimate of drug-likeness (QED) and synthetic accessibility score (SAS) of the molecules. A compound with higher QED is more drug-like, while lower SAS means it can be more easily synthesized. The other plugin was trained on the ExCAPE-DB dataset with an interaction predictor to predict the binding to a set of 3 related genes. We used a weighted sum of the desired molecule properties and target binding affinities as an objective function.

We were able to generate hundreds of novel and valid compounds which were not presented in any of the three datasets. We tested promising candidates with high fitness scores and made interaction predictions with the Swiss Target Prediction [17] software, and the molecules showed high binding affinity to the selected targets.

While the average target prediction accuracy did not improve significantly from the model trained without the plugin architecture, but the additional benefit of using an arbitrary number of plugins trained for different tasks allows rapid customization of desired target profiles, compared to retraining the entire network.

IV. Conclusion and Future Work

We have seen the challenges inherent in drug discovery and that the combination of generative and predictive models to reach a selective binding profile plays an increasingly important role in this field of research.

In our paper we introduced a method which can be used to generate suitable candidates for an arbitrary objective function. By separating the task of reconstruction and conditioned generation we end up with a model capable to efficiently handle newly arriving targets. Finally, we shoved how it can be used in a distributed environment without sharing any confidential data.

For further improving the model we want to try the universal transformer architecture [19] and the relative position embedding [20].

References

[1] S. Kim et al., "PubChem substance and compound databases", Nucleic acids research, vol 44, no D1, bll D1202–D1213, 2016.

[2] P. G. Polishchuk, T. I. Madzhidov, en A. Varnek, "Estimation of the size of drug-like chemical space based on GDB-17 data", Journal of computer-aided molecular design, vol 27, no 8, bll 675–679, 2013.

[3] R. Gómez-Bombarelli et al., "Automatic chemical design using a data-driven continuous representation of molecules", ACS central science, vol 4, no 2, bll 268–276, 2018.

[4] W. Jin, R. Barzilay, en T. Jaakkola, "Junction tree variational autoencoder for molecular graph generation", in International conference on machine learning, 2018, bll 2323–2332.

[5] Z. Zhou, S. Kearnes, L. Li, R. N. Zare, en P. Riley, "Optimization of molecules via deep reinforcement learning", Scientific reports, vol 9, no 1, bll 1–10, 2019.

[6] Ł. Maziarka, A. Pocha, J. Kaczmarczyk, K. Rataj, T. Danel, en M. Warchoł, "Mol-CycleGAN: a generative model for molecular optimization", Journal of Cheminformatics, vol 12, no 1, bll 1–18, 2020.

[7] A. Vaswani et al., "Attention is all you need", in Advances in neural information processing systems, 2017, bll 5998–6008.

[8] R. Irwin, S. Dimitriadis, J. He, en E. J. Bjerrum, "Chemformer: A Pre-Trained Transformer for Computational Chemistry", Machine Learning: Science and Technology, 2021.

[9] N. Brown, M. Fiscato, M. H. S. Segler, en A. C. Vaucher, "GuacaMol: benchmarking models for de novo molecular design", Journal of chemical information and modeling, vol 59, no 3, bll 1096–1108, 2019.

[10] H. Öztürk, A. Özgür, en E. Ozkirimli, "DeepDTA: deep drug–target binding affinity prediction", Bioinformatics, vol 34, no 17, bll i821–i829, 2018.

[11] X. Tan et al., "Automated design and optimization of multitarget schizophrenia drug candidates by deep learning", European Journal of Medicinal Chemistry, vol 204, bl 112572, 2020.

[12] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, en B. Frey, "Adversarial autoencoders", arXiv preprint arXiv:1511. 05644, 2015.

[13] L. Fang, T. Zeng, C. Liu, L. Bo, W. Dong, en C. Chen, "Transformer-based Conditional Variational Autoencoder for Controllable Story Generation", arXiv preprint arXiv:2101. 00828, 2021.

[14] J. Sun et al., "ExCAPE-DB: an integrated large scale dataset facilitating Big Data analysis in chemogenomics", Journal of cheminformatics, vol 9, no 1, bll 1–9, 2017.

[15] Y. Duan, C. Xu, J. Pei, J. Han, en C. Li, "Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders", arXiv preprint arXiv:1911. 03882, 2019.

[16] T. Yang et al., "DrugSpaceX: a large screenable and synthetically tractable database extending drug space", Nucleic acids research, vol 49, no D1, bll D1170–D1178, 2021.

[17] A. Daina, O. Michielin, en V. Zoete, "SwissTargetPrediction: updated data and new features for efficient prediction of protein targets of small molecules", Nucleic acids research, vol 47, no W1, bll W357–W364, 2019.

[18] L. Lyu, H. Yu, en Q. Yang, "Threats to Federated Learning: A Survey", arXiv [cs.CR]. 2020.

[19] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, en Ł. Kaiser, "Universal transformers", arXiv preprint arXiv:1807. 03819, 2018.

[20] P. Shaw, J. Uszkoreit, en A. Vaswani, "Self-attention with relative position representations", arXiv preprint arXiv:1803. 02155, 2018.