Exploring the Effects of Pretraining Encoders on DTI Prediction

Raul Sofia[1][2019225303] and Paulo Correia[1][2020226382]

Department of Informatics, University of Coimbra
raulsofia,paulocorreia@student.dei.uc.pt
https://www.uc.pt/fctuc/dei/

Abstract. Drug target interaction (DTI) prediction plays a crucial role in drug discovery. However, predicting DTIs accurately is challenging due to the complex and multifaceted nature of chemical structures and biological interactions. Recurrent neural network (RNN) autoencoders have emerged as a promising approach for learning efficient representations of molecules, which can be leveraged for DTI prediction. This study investigates the use of molecule RNN autoencoders for learning representations from both molecular and target structures to predict DTIs. The proposed method, is trained on a dataset of molecules, where the RNN autoencoder captures the intrinsic structure of each molecule.

Keywords: DTI · Encoder-Decoder · RNN.

1 Introduction

The pursuit of novel therapeutic agents in drug discovery and development relies extensively on deciphering the intricate landscape of Drug-Target Interactions (DTIs). Accurate prediction of these interactions remains a formidable challenge due to the complex and multifaceted nature of chemical structures and biological processes. In recent years, recurrent neural network (RNN) autoencoders have emerged as a promising paradigm for learning efficient representations of molecules, offering a transformative approach to decoding the language of chemical interactions [3, 4].

This study represents a systematic exploration into the application of RNN autoencoders to glean meaningful representations from both molecular and target structures, with the ultimate goal of enhancing DTI prediction accuracy. The central hypothesis underpinning this investigation posits that leveraging autoencoders for pretraining on a diverse and extensive dataset can yield more generalized molecular representations, thereby addressing the challenges inherent in accurate DTI prediction.

The methodological framework of this study involves the training of a dedicated model on a curated dataset of molecules, where the RNN autoencoder captures the intrinsic structural features of each molecule. Emphasis is placed on understanding the impact of pretraining encoders on DTI prediction, aiming to unravel the efficacy of this approach in navigating the intricacies of drug discovery.

To fortify the empirical foundation of our investigation, we draw upon the ZINC15 dataset, a comprehensive repository housing 220 million molecules [8]. Within this expansive collection, a judicious subset of 250,000 molecules is meticulously chosen to ensure a focused exploration, based on the work of [3], that accomplishes a similar task to the one we propose here. Additionally, the study incorporates the Adenosine A2A receptor Affinity Dataset, encompassing binding affinity information for 4,534 distinct molecular entities. While recognizing the value of affinity data, we acknowledge its potential limitations and posit that the incorporation of RNN autoencoders for pretraining on a more diverse dataset may enrich the predictive capacity of our model.

In essence, this study seeks to contribute to the evolution of DTI prediction methodologies by exploring the untapped potential of RNN autoencoders and pretraining encoders. Through a nuanced investigation using both the ZINC15 and Adenosine A2A receptor Affinity Dataset, we endeavor to provide valuable insights that advance the accuracy and efficacy of DTI prediction within the complex domain of drug discovery.

2 Materials and Methods

2.1 Datasets

ZINC The ZINC15 dataset, a comprehensive repository in the domain of drug discovery and computational chemistry, encompasses a rich collection of chemical compounds meticulously curated to support various scientific endeavors. With a repository comprising 220 million molecules, this resource is particularly well-suited for applications in high-throughput virtual screening and structure-based drug design.

In the specific context of our study, a judicious subset of the extensive ZINC15 dataset was employed, focusing on 250000 molecules. This deliberate selection allows for a more manageable exploration within the larger dataset, ensuring a nuanced and insightful analysis.

Adenosine A2A receptor Affinity Dataset The dataset utilized in this study was systematically collated from scholarly sources, specifically derived from [7]. It encompasses the binding affinity information of 4534 distinct molecular entities with the Adenosine A2A receptor, quantified through the pCHEMBL score. While the original dataset included logP and molecular weight values, these parameters were deemed non-contributory to the objectives of our investigation and consequently excluded from further analysis. With our approach we managed to achieve better results then simple training a new model and even surpassing the architectures found in the literature.

SMILES Linear representation formats are a class of symbolic systems characterized by a coherent sequence of symbols adhering to specific syntactical rules, sharing notable similarities with natural language structures. In contrast

to more intricate representations, these formats aim to streamline complexity by discarding structural intricacies and concealing implicit information. Despite their redundancy, their non-ambiguous nature renders them well-suited for targeted applications.

In the context of this study, the chosen linear representation notation is SMILES (Simplified Molecular Input Line Entry System) [5], recognized as one of the most widely adopted formats in the scientific community. SMILES represents individual atoms through their respective atomic symbols, such as 'C' for carbon and 'N' for nitrogen. The connectivity between atoms is established by bond characters, wherein simple bonds are typically implicit but can be denoted by the '-' character. Additionally, a double bond is symbolized by '=', while a triple bond is represented by '#'. Aromaticity is conveyed through lowercase letters or the symbol ':' to signify the presence of aromatic bonds. Hydrogen atoms, while capable of being explicitly denoted by the letter 'H', are frequently omitted for brevity.

This chosen linear representation, namely SMILES, facilitates the concise and unambiguous portrayal of molecular structures, aligning with the overarching objective of minimizing representation complexity for the intended analytical task, and is widely used and validated across DTI literature [3, 4, 7, 9].

Preprocessing Both datasets had some transformations applied. However, SMILES preprocessing was common to both. Fisrt of all, we removed any information present in the SMILES string that was not related to the chemical properties relevant for this work. This includes the removal of any labels associated to atoms (e.g. atom numeration), information about isotopic forms of atoms (e.g. ^{13}C), as mass number does not affect greatly the chemical properties of the molecules, and radioactive isotopes are frequently present in datasets simply due to their usage as markers in experiments [1]. There are, however, recent studies that shed some light over the manipulation of mass number on atoms in drugs, especially around lightly reducing their potency with heavier, slower compounds [6]. We still considered that the noise introduced in tokenization wouldn't worth their inclusion. We also removed any associated salts. Arguably, this last one could have some importance in the drug effectiveness, due to the fact that the salt form of a drug can affect its solubility and bioavailability [2]. However, we intend to focus solely on the drug-target interaction, and therefore we ignore bioavailability and remove any salts associated to the drug, keeping only the ligand itself. Note, however, that stereochemistry is preserved, as this is of critical importance for the molecular geometry and therefore the interactions studied here.

Next, we applied a tokenization process to the SMILES strings. This process consists of splitting the SMILES string into its constituent parts, which are then mapped to a vocabulary of tokens. This vocabulary was derived from the large ZINC dataset and is composed of not only the individual elements (organic and heteroatoms, both normal and aromatic forms) (e.g. C, c, N, O, etc.), but also of some functional groups (e.g. -NH2, -NH3, -OH, etc.), the stereocenters (e.g.

[C@], [C@@], etc.), the general SMILES notation punctuation (e.g. -, =, #, etc.), and a starting and ending/padding token ('G' and 'A', respectively). There are 83 tokens in total. Tokenization starts with splitting the string in the relevant tokens (in a greedy strategy, largest tokens first), then prepending a start token and appending one or more padding tokens until the a maximum number is met. Here, the maximum number of tokens was set to 100, after analysis of the datasets. In the rare ocasion one of the SMILES strings was larger than 100 tokens, its was removed from the dataset (truncation would originate mostly invalid molecules, and certainly with very different chemical properties). The resulting tokenized SMILES strings were then mapped to integers.

After these common steps, no further transformations were applied to the SMILES, except for their encoding as one-hot vector to serve as targets to the autoencoder. We further processed the A2A dataset. The pCHEMBL value, used as a target for the predictor, was normalized to a range of 0 to 1. This was done by subtracting the minimum value and dividing by the maximum value.

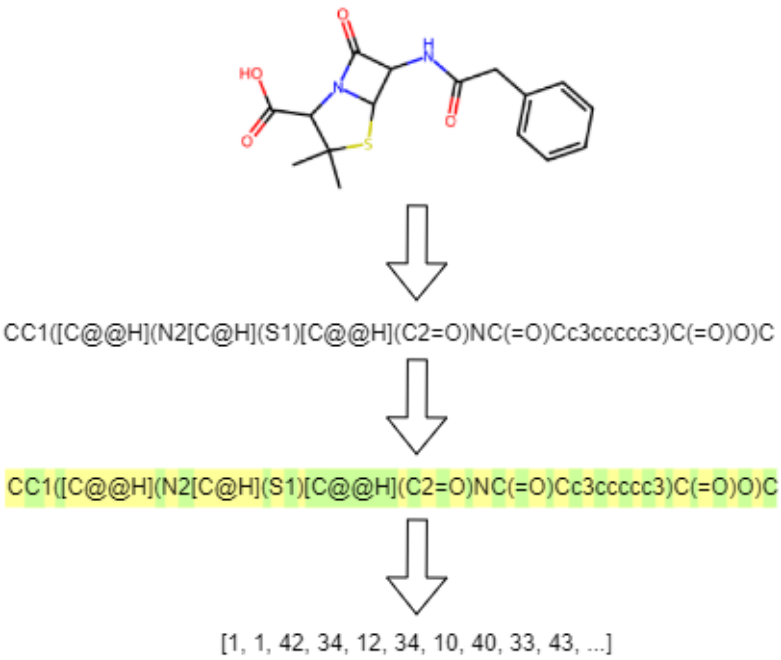

Fig. 1. Tokenization procedure.

pCHEMBL The pCHEMBL value is a measure of the binding affinity of a drug to a target. It is defined as the negative logarithm of one of the most common binding affinity metrics, such as the dissociation constant (Kd), the

inhibition constant (Ki), or the half maximal inhibitory concentration (IC50), among others. Although somewhat arbitrary (due to the differences between the underlying metrics), it has been widely validated by literature, while allowing for bigger datasets, and therefore will be used here too.

2.2 Models

Autoencoder To work around the data scarcity of this work while learn efficient representations over a more generalized set of molecules, a simple autoencoder architecture was devised. The encoder consisted of an Embedding layer that projected the uni-dimensional vector returned in the tokenizing and encoding steps, correspondent to each token, into an embedding vector with 128 dimensions. Two GRU layers with 512 units and return_sequences set to true followed, with an output layer with units corresponding to the encoding dimension selected for the autoencoder and return_sequences set to false. The encoder therefore returns a vector with a size equal to the encoding dimensions defined.

To effectively train the encoder, an decoder model is needed that can be discarded at the end as it is not useful for our work. This model consisted of 3 GRU layers with 512 units and a final timedistributed dense with a softmax activation.

$$L(\theta, \phi) := E_{x \sim \mu_{ref}}[d(x, D_\theta(E_\phi(x)))] \tag{1}$$

The reconstruction loss of a typical autoencoder is defined in equation (1), where in this case d, was defined as the categorical crossentropy between the one-hot encoded representation of the initial sequence and the output of the decoder model. The model was trained on the ZINC subset with 250k molecules until either reaching 10 epochs or the early stopping with a patience of 5 epochs monitoring the validation loss halted the training.

Predictor The predictor used in this work is a simple feed-forward fully connected neural network, with 2 hidden layers of 512 and 256 neurons, respectively, and a final output layer with a single neuron. The activation function used in the hidden layers was the rectified linear unit (ReLU), while the output layer used a sigmoid activation function. The loss function used was the mean squared error (MSE), and the optimizer was the Adam optimizer with default hyperparameters. The model was trained for 100 epochs with a batch size of 32 but, again, early stopping was used, with a patience of 10 epochs. To prevent overfitting, a dropout of 0.4 (we noted the model was overfitting a lot) was introduced between every two layers. The data input were the encodings resulting from the encoder layers, and the targets were the normalized pCHEMBL values, as described previously.

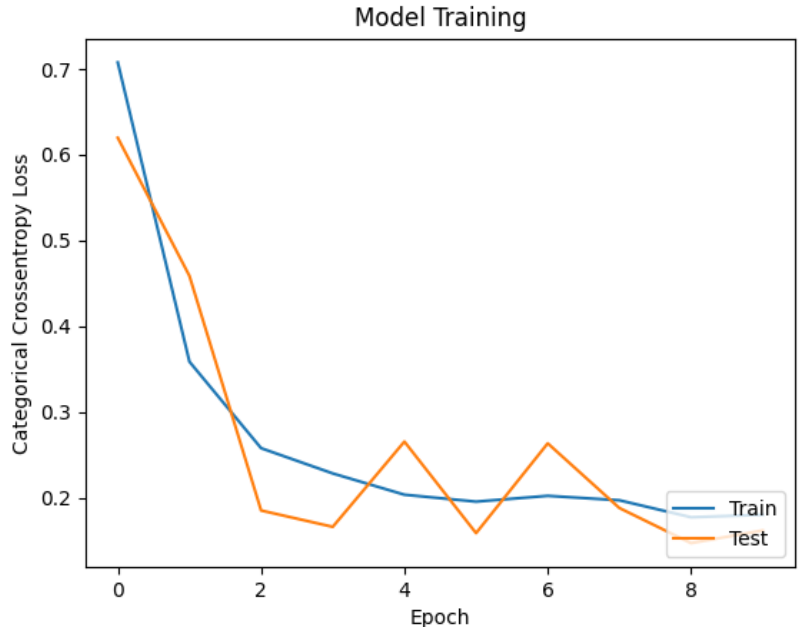

Fig. 2. Training progress of the autoencoder with latent dimension of 128.

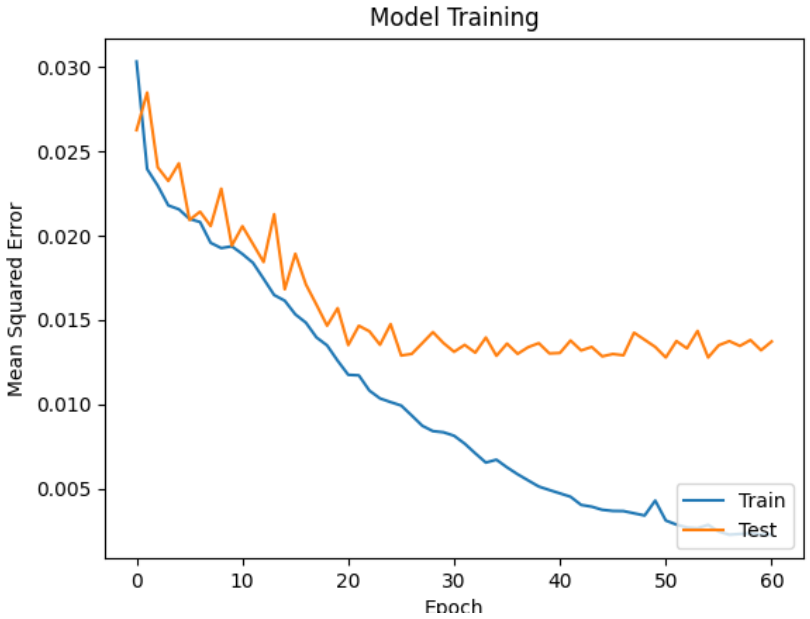

Fig. 3. Training progress of the predictor with latent dimension of 128 and pre-trained non-freezed encoder.

3 Experiments and Results

3.1 Training

Autoencoder The figure 2 shows a very typical training curve where the validation loss showed some variation but kept its values close to the training loss. As so, the early stopping did not halt the process. The model ended with a final loss of 0.1801 and a validation loss of 0.1618. We can conclude from this data that the model did converge to create a good representation.

Predictor The figure 3 shows a training curve where the validation loss accompanied the training loss until the 0.015 mark, where it stagnated. As so, the early stopping halted the process at the 60th epoch. The model ended with a final loss of 0.00310 and a validation loss of 0.01278. We can conclude from this data that the model had some overfitting issues, making it difficult to reduce the validation loss.

3.2 Varying the encoding dimension of the encoder

To test the effect of the encoding dimension of in the performance of the predictor three encoding models were used only varying the size of the encoding dimension. We choose the following sizes: 128, 64 and 32.

Table 1. Mean Squared Error achieved for each of the different encoding dimensions.

Encoding dimension	MSE
128	**0.0110**
64	0.0144
32	0.0153

As expected, higher encoding dimensions lead to better results in the prediction task, this effect might be an expression of several effects, mainly the reduction of the bottleneck, allowing to pass-through information in an unchanged manner and the simple increase in model size resulting from the higher amount of parameters. However, if a plateau performance was achieved after a certain dimentionality, we would have picked the smallest. As no such thing was noted, we opted for the best performant.

3.3 Fine-tuning: Freezing weights vs Retraining vs Reinitializing model

In this experiment, three strategies were adopted in order to fine-tune the model. The first one was to freeze the weights of the encoder and only train the predictor (the main goal of this study). The second one was to retrain the whole model,

encoder and predictor, starting with the pretrained weights (the encoder weights were not frozen). The third one was to reinitialize the whole model, encoder and predictor, and train it from scratch. The last two strategies were used as a baseline, to compare against the performance of the first one. The rationale behind this is that the encoder weights, being pretrained, should already be close to the optimal weights, and therefore should not need much change. The predictor, on the other hand, should be able to learn the task starting from the encoded representation of the molecules, and therefore, ideally, should not need to change the pretrained weights. The third strategy was used to assess the importance of the pretraining over a larger dataset (ZINC) in the performance of the model.

We tested these strategies on best performing encoder model, using 128-dimentional latent space. We also compared them against the LSTM architecture proposed by [7]. The results obtained were the following:

Table 2. Mean Squared Error, Concordance Correlation Coefficient and Pearson's Squared Correlation Coefficient achieved for each of the different strategies, obtained by comparing the predictions over the test set to the real values.

Strategy	**MSE**	**CCC**	r^2
Freezing	0.0163	0.5989	0.4502
Retraining	**0.0110**	**0.7811**	**0.6255**
Reinitializing	0.0115	0.7643	0.6204
LSTM predictor	0.0128	0.7373	0.5366

As we can see, the best performing strategy was to retrain the whole model, encoder and predictor, starting with the pretrained weights. This is in line with our expectations, as the encoder weights should already be close to the optimal weights. We didn't expect such a poor performance from the freezing strategy (the classical transfer learning approach), but this might arise from the fact that the encoder was trained on a very different dataset, and therefore the encoded representation might not be that optimal for the task at hand. It might be that the encoded representations eliminate some of the information relevant for the task, as it was not explicitly trained for it. The reinitializing strategy, on the other hand, performed very well, almost as well as the retraining strategy. This was more or less expected, as they share the exact same architecture. The small difference may come from the already lightly optimized weights of the encoder in the pretraining case (the reinitializing strategy starts with random weights). The pretrained network already produced a reasonable encoded representation of the molecules, but, differently from the freezing case, the encoder and the predictor were allowed to co-adapt for the new task, achieving the best performance. As a comparison to other existing models, we also trained a LSTM predictor from [7] on the same dataset. This network, although with similar layers to the networks

used here, is considerably less complex, and therefore the poorer performance is not surprising. Nevertheless, it is still good to note the improvement.

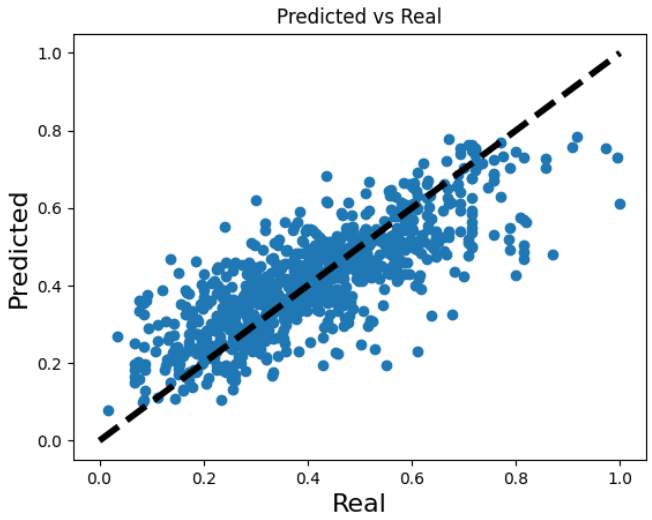

Fig. 4. Comparison of predicted and real values for the predictor with latent dimension of 128 and pre-trained non-freezed encoder.

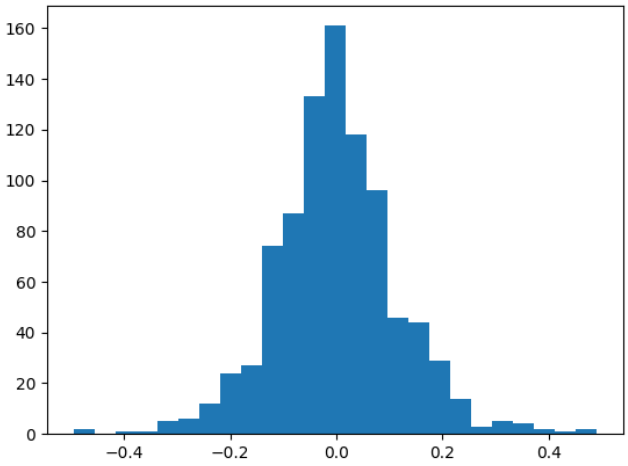

Fig. 5. Error for the predictor with latent dimension of 128 and pre-trained non-freezed encoder.

For the comparison of the predicted vs the real values, we see that the predictions tend to align with the expected value, approaching the diagonal line. This

indicates that the model gives predictions that approach the expected values, without diverging too much.

As for the average error, the deviations displayed a normal distribution centered on 0 (0.0012) and a standard deviation of 0.1172. This indicates that, on average the results did not deviate unequally to one side or the other.

4 Conclusions

In conclusion, the study on exploring the effects of pretraining encoders on DTI prediction has yielded promising results. The proposed method, which leverages RNN autoencoders for pretraining on a diverse and extensive dataset, has shown to yield more generalized molecular representations, thereby addressing the challenges inherent in accurate DTI prediction. The best performing strategy was to retrain the whole model, encoder, and predictor, starting with the pretrained weights. The study also highlights the overfitting issues faced by the model, making it difficult to reduce the validation loss. Nonetheless, the research provides valuable insights into the potential of RNN autoencoders in advancing the predictive capacity of models in the realm of drug discovery and development. The incorporation of diverse datasets and the judicious selection of molecules contribute to enriching the empirical foundation of the research, paving the way for future advancements in this critical domain.

References

1. Charles S. Elmore and Ryan A. Bragg. Isotope chemistry; a useful tool in the drug discovery arsenal. *Bioorganic Medicinal Chemistry Letters*, 25:167–171, 1 2015.
2. Deepak Gupta, Deepak Bhatia, Vivek Dave, Vijaykumar Sutariya, and Sheeba Varghese Gupta. Salts of therapeutic agents: Chemical, physicochemical, and biological considerations. *Molecules : A Journal of Synthetic Chemistry and Natural Product Chemistry*, 23, 2018.
3. Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4:268–276, 2 2018.
4. Qiwan Hu, Mudong Feng, Luhua Lai, and Jianfeng Pei. Prediction of Drug-Likeness Using Deep Autoencoder Neural Networks. *Frontiers in Genetics*, 9:422486, nov 2018.
5. Qiwan Hu, Mudong Feng, Luhua Lai, and Jianfeng Pei. Prediction of Drug-Likeness Using Deep Autoencoder Neural Networks. *Frontiers in Genetics*, 9:422486, nov 2018.
6. Rita Maria Concetta Di Martino, Brad D. Maxwell, and Tracey Pirali. Deuterium in drug discovery: progress, opportunities and challenges. *Nature Reviews Drug Discovery 2023 22:7*, 22:562–584, 6 2023.
7. Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P. Arrais. Diversity oriented deep reinforcement learning for targeted molecule generation. *Journal of Cheminformatics*, 13:1–17, 12 2021.

8. Teague Sterling and John J. Irwin. Zinc 15 - ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55:2324–2337, 11 2015.
9. Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 09 2018.

5 Supplementary Data

All the code, results and datasets used along this project can be accessed via `https://github.com/RaulSofia/Bioinf_proj`