CS230

A Generative Deep Learning Approach for Alzheimer's Disease Drug Discovery

Justin Shen (Jshen3), Akash Gupta (apgupta3), Adrian Gamarra Lafuente (a1090588)

1.0 Introduction

Current drug discovery efforts revolve around a time-intensive timeline that can take 10-15 years, oftentimes longer for complex neurological diseases like Alzheimer's disease [1]. This is a direct result of the large chemical space (approximately 10^{60} possible drug-like molecules) to be explored [2]. In current drug discovery procedures, candidate compounds are oftentimes empirically selected from the general chemical space, an inefficient approach.

With the development of deep learning as an efficient method to process large amounts of data, recent research has proposed the use of deep learning approaches to accelerate the drug discovery process. Specifically, new research has demonstrated the potential for generative deep learning approaches in novel chemical generation using natural language processing (NLP) methodologies [3].

2.0 Related Work

In 2018, Gupta et al. investigated a long short-term memory (LSTM) recurrent neural network (RNN) approach for drug discovery. Their research concluded that LSTM RNNs hold a tremendous amount of promise for computer-aided chemical design, as their approach was able to generate structurally similar molecules to their training set. They were also able to leverage their model to "grow" new compounds from chemical fragments, useful for finding chemical derivatives to compounds of interest [3].

Similarly, Segler et al. also highlighted the potential of LSTM RNNs in generating focused compound libraries with a series of chosen properties. In their research, they targeted the 5-HT_{2A} receptor as a target of interest. They trained their LSTM RNN model on a generic, unfocused library of chemical strings to allow their model to learn generic chemical representation. Next, they employed transfer learning and retrained their model on a focused dataset of previously determined 5-HT_{2A} receptor inhibitors. Their results indicate that the generated compounds from this model were structurally similar to the previously determined 5-HT_{2A} receptor inhibitors, but were chemically unique, thus generating novel chemicals for 5-HT_{2A} receptor inhibition. Using a classification machine learning model to predict the inhibitory potential of the generated compounds, over 60% of the generated compounds were predicted to inhibit the 5-HT_{2A} receptor (20 epochs of fine tuning) [4].

In our research, we'll target β-secretase 1 (BACE-1), a pivotal protein in Alzheimer's disease progression.

3.0 Dataset & Features

3.1 Data Overview

Gupta et al. and Segler et al. demonstrated the promise of transfer learning in chemical generation (first training on a generic chemical dataset and then training on a focused chemical dataset). As a result, our training methodology will consist of training our models on both a generic chemical dataset, allowing the model to learn nonspecific chemical syntax, and a focused chemical dataset, allowing the model to learn more complex chemical features specific to Alzheimer's disease and BACE-1 [3,4].

Our generic chemical dataset consisting of 300,000 chemical compounds was sourced from ChEMBL, a publicly-accessible database of compounds for virtual screening (Figure 1A) [5].

Our focused chemical dataset consisting of 1560 inhibitors for BACE-1 was also sourced from ChEMBL, a curated chemical database of proteins and their inhibitors (Figure 1B) [5].

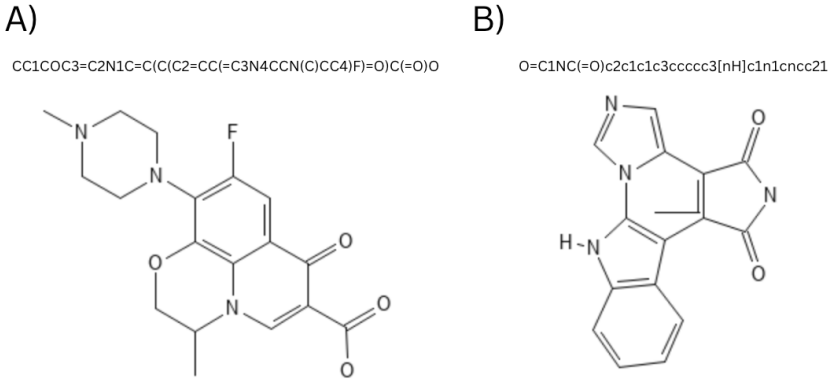

Figure 1. A) A molecule from our general dataset. B) A molecule from our Alzheimer's dataset.

3.2 Data Preprocessing

We split our dataset into training (72%) , testing (20%), and validation sets (8%), with percentages representing the data breakdown. We then processed all the datasets by fitting the Keras tokenizer to the corresponding data sets, allowing us to convert the raw characters into model-usable numerical values.

3.3 Feature Generation

In order to convert our raw character strings into data features learnable by our deep learning models, we leveraged a modified form of n-grams sequence processing to allow us to predict for the next character that follows a 15 character input sequence (Figure 2). In this feature representation, the model takes in the last 15 characters to predict the next character, allowing us to sequentially generate text.

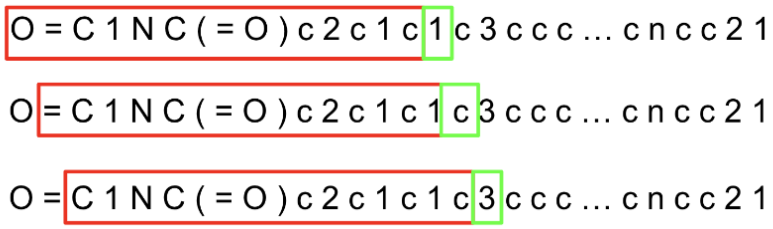

Figure 2. Modified N-grams Feature Generation

4.0 Methods

4.1 Model Overview

In our project, we decided to implement three models from scratch in order to rigorously test our hypothesis. First, we implemented our own baseline model (Model A) consisting of one LSTM layer of 32 units and one dropout layer with a probability of 0.2 (Figure 3A). This baseline not only offered a point of comparison but gave us a foundation to work off of for our more complex models. For our novel model (Model B), we combine a sequence of gated recurrent unit (GRU) layers, with 128 units, and a sequence of LSTM layers consisting of 512 units(Figure 3B). We explored the efficacy of this unique combination because previous papers have highlighted the use case of both GRU and LSTM layers for molecular generation, but none explored their use case in conjunction [3,4,6]. Our third model (Model C) consisted of only LSTM layers with 512 units, an was a more complex version of our baseline model with five LSTM layers (Figure 3C). All of our models had the same Dense output layer of 43 units to represent the 43 output possibilities of our dataset.

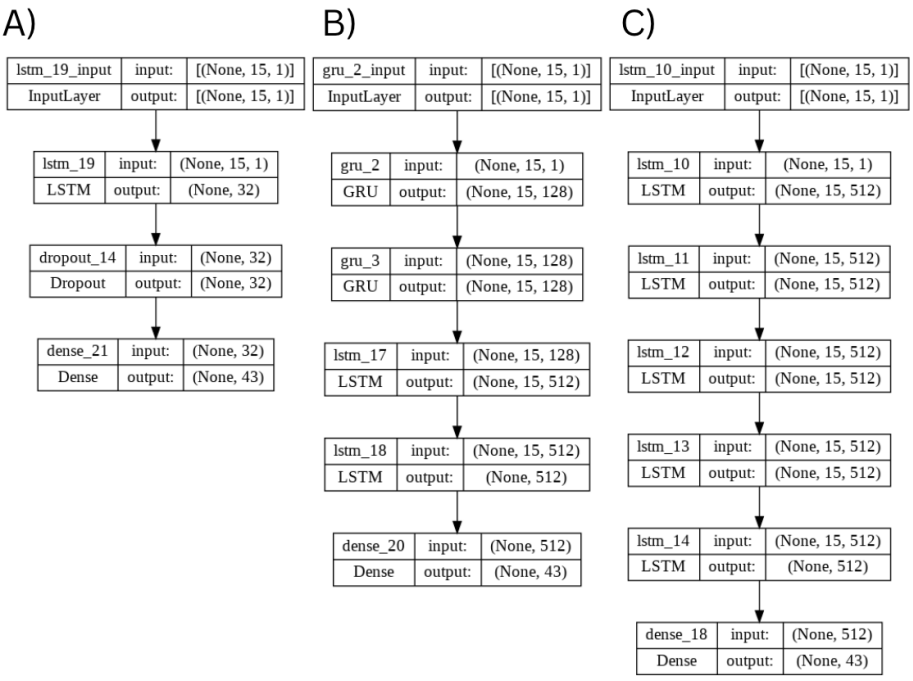

Figure 3. A) Baseline architecture. B) Hybrid architecture. C) Improved baseline architecture

4.2 Model Training & Transfer Learning

To improve our model's learning capabilities, we leveraged transfer learning to train our three models, first allowing our model to learn from the generic dataset and then fine tuning its weights on the specific dataset for Alzheimer's disease. We trained our models for 40 epochs, but implemented early stopping with a patience of five epochs to ensure our model accuracy and loss metrics didn't deteriorate over time.

4.3 Hyperparameter Tuning

In order to optimize our model's hyperparameters, we leveraged Bayesian optimization, which specializes in an informed search of a model's hyperparameters. Unlike other optimization protocols like random search, Bayesian leverages knowledge from previous trials to make an educated decision when deciding the next trial's optimal parameters, which significantly speeds up the optimization time, while providing better results [7]. For Model B and Model C, we optimized for their parameters using Bayesian optimization for three epochs and three trials for each model. We then used the results from these trials to dictate our hyperparameter tuning.

5.0 Results/Discussion

5.1 Model Metrics

Table 1. Model Test Metrics on General Dataset (Before Transfer Learning)

General Test Metrics	*Baseline Model (Model A)*		*Hybrid Model (Model B)*		*Improved Baseline Model (Model C)*	
	Accuracy	*Loss*	*Accuracy*	*Loss*	*Accuracy*	*Loss*
Results	*65.5%*	*1.193*	*76.7%*	*0.788*	*77.4%*	*0.770*

On our generic data, prior to transfer learning, we evaluated our three models' abilities to learn nonspecific chemical syntax from our generic test dataset. Our baseline model (Model A) achieved an accuracy of 65.5% while the loss

remained at 1.193. Using this as our point of reference for developing future models, we then evaluated our hybrid model (Model B) on the same test dataset, achieving an accuracy of 76.7% and a loss of 0.788. Lastly, we evaluated our improved baseline model (Model C) and achieved an accuracy of 77.4% and a loss of 0.770 (Table 1).

Table 2. Model Test Metrics on Alzheimer's Disease Dataset (Before Transfer Learning)

Focused Test Metrics	*Baseline Model (Model A)*		*Hybrid Model (Model B)*		*Improved Baseline Model (Model C)*	
	Accuracy	*Loss*	*Accuracy*	*Loss*	*Accuracy*	*Loss*
Results	*56.0%*	*1.476*	*88.0%*	*0.468*	*88.1%*	*0.473*

After transfer learning, we evaluated our three models' abilities to learn specific chemical features specific to Alzheimer's disease from our Alzheimer's test dataset. Our baseline model (Model A) achieved an accuracy of 56.0% while the loss remained at 1.476. Using this as our point of reference for developing future models, we then evaluated our hybrid model (Model B) on the same test dataset, achieving an accuracy of 88.0% and a loss of 0.468. Lastly, we evaluated our improved baseline model (Model C) and achieved an accuracy of 88.1% and a loss of 0.473 (Table 2).

As seen in the results above, we were able to improve our LSTM model from a 56.0% accuracy in Model A to 88.1% accuracy in Model C, due to the error analysis. Originally our baseline model (Model A) had a training accuracy around 50.7% and a validation accuracy around 55.2%, which implied that we had high bias. To combat this, we knew we had to make a bigger network or train longer. Since our accuracy had already flattened out after around 20 epochs even though the model was trained for 40 epochs, we decided the better solution would be to make a bigger network. Therefore, we made our improved model (Model C) to have 5 LSTM layers. This change in our model architecture, along with the tuning of the hyperparameters, allowed us to achieve a training accuracy of 94.2, and a testing accuracy of 88.1%, a significant improvement over our baseline model testing accuracy of 56.0%.

Additionally, we wanted to try a novel architecture for molecular generation. During our literature review, we found that past papers had implemented only LSTM models and only GRU models, but we wanted to try combining both types of models to see if we could reap the benefits of both types of models [3,4,6]. The GRU model is computationally faster and works better with short sequences, while the LSTM model is generally more accurate with longer sequences. Keeping this in mind, we strove to achieve the fast computational efficiency of the GRU on the early layers that learn the simpler features of the chemical sequence, while also keeping the high accuracy of the LSTM model when learning the more complex features [8]. Based on the results above, our hybrid model performed just as well as our LSTM model and was able to train much faster, making it computationally more efficient.

5.2 Molecule Generation

Using our improved baseline model (Model C), we generated 18 new molecules using randomly selected fragments from our Alzheimer's dataset. We drew two of these molecules into chemical structures as examples (Figure 4).

As shown, our molecules are slightly smaller than the reference molecules in the dataset, this is believed to be a result of our coded stop character ('!') in our dataset that stops the model from generating new chemicals once the stop character is predicted. Despite this, we were still able to generate valid chemicals from molecular fragments.

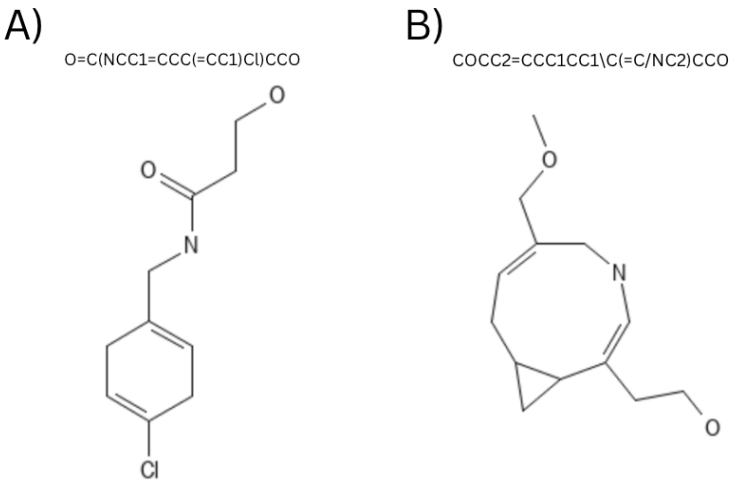

Figure 4. A) Generated molecule 1. B) Generated molecule 2.

Conclusions/Future Work

This work explored the viability of recurrent neural networks in drug generation for Alzheimer's disease. We trained three models, a baseline model, a novel hybrid model, and an improved baseline model. Our results significantly improved over our baseline model (Model A), which allowed us to be able to generate 18 new molecules that can help inhibit BACE-1, slowing down Alzheimer's disease. In the end, we hope to continue working on these models past this project to see if we can improve our models and generate more molecules, so that we can help the many families struggling from this painful disease. Going forward, we could build off our project by exploring implementing a variational autoencoder (VAE) architecture that can add more variability to our generated results. Additionally, we can perform more rounds of error analysis on our improved baseline (Model C) and hybrid models (Model B) to try to increase our accuracy on the test data. Finally, we look forward to trying to validate that the molecules we generated are functional for Alzheimer's disease treatment, so that we can see how our work could get implemented to help people in real life.

Contributions

Justin Shen did a lot of the literature overview of the project. Justin worked on pre-processing the data so it is usable by our model. Moreover, Justin wrote code to generate new molecules and conducted the Bayesian Optimization to tune our hyperparameters

Adrian Gamarra Lafuente has been responsible for a lot of the coding and training of models on the generated dataset. Adrian has contributed to data pre-processing and the initial design of the baseline model with the Keras API. Adrian has also been working with Justin on planning further implementation of the generative model. Adrian set up the AWS computation and training, as well as the source code management.

Akash Gupta has been responsible for performing the error analysis on the baseline model to help design our improved model. Akash also worked on the transfer learning of all the models and evaluating the models that we built based on test accuracy.

Bibliography

[1] J. Cummings, P. S. Aisen, B. DuBois, L. Frölich, C. R. Jack, R. W. Jones, J. C. Morris, J. Raskin, S. A. Dowsett, and P. Scheltens, “Drug development in alzheimer’s disease: The path to 2025,” Alzheimer's Research & Therapy, vol. 8, no. 1, 2016.

[2] J.-L. Reymond, “The Chemical Space Project,” Accounts of Chemical Research, vol. 48, no. 3, pp. 722–730, 2015.

[3] 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, 2017.

[4] M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, “Generating focused molecule libraries for drug discovery with recurrent neural networks,” ACS Central Science, vol. 4, no. 1, pp. 120–131, 2017.

[5] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. P. Overington, “Chembl: A large-scale bioactivity database for Drug Discovery,” Nucleic Acids Research, vol. 40, no. D1, 2011.

[6] S. Huang, H. Mei, L. Lu, M. Qiu, X. Liang, L. Xu, Z. Kuang, Y. Heng, and X. Pan, “De novo molecular design of caspase-6 inhibitors by a GRU-based recurrent neural network combined with a transfer learning approach,” Pharmaceuticals, vol. 14, no. 12, p. 1249, 2021.

[7] T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh, “Fast hyperparameter tuning using bayesian optimization with directional derivatives,” Knowledge-Based Systems, vol. 205, p. 106247, 2020.

[8] S. Yang, X. Yu, and Y. Zhou, “LSTM and GRU neural network performance comparison study: Taking Yelp Review dataset as an example,” 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), 2020.