Implementing a deep-neural network (DNN) based model to aid drug discovery

Krishna D.B. Anapindi

1. Abstract

Quantitative Structure-Activity Relationship (QSAR) studies are performed to identify the lead drug molecules from several hundred thousand candidates. Since the beginning of the 21st century, computational methods such as support vector machines (SVM), random forests (RF) and Neural Networks (NN) were widely used to predict the activity of these potential drug candidates. These approaches saved both time and money that went into the traditional approaches of drug screening. However, with the increasing size of data used as input for these computational models, there is a need to come up with efficient approaches for feature selection that can be implemented to both speed up the model evaluation and provide better results. Here I propose a genetic selection-based algorithm for feature selection followed by a grid search analysis to find the optimal parameters that work best on the reduced feature space. Finally, I compare these results to the current state-of the art Deep Neural Network (DNN) based approach for predicting molecular activity.

2. Introduction

During a drug discovery process, the efficacy of a potential drug-candidate is estimated by screening it against the target molecule and other peripheral off-target entities. Ideally, a candidate that shows high potency for the target molecule and low/no potency towards an off-target molecule is highly desirable. In a typical drug discovery pipeline, several hundred thousand molecules are synthesized and then screened against the desired targets to evaluate their potential as a drug candidate. These studies are called the quantitative structure-activity relationship (QSAR) studies that can quantitatively predict the relationship between the structure of a molecule and its biological activity towards the target and off-target molecules. However, QSAR studies typically involve over 100,000 compounds each with several thousand descriptors. Hence, evaluating their efficacy by synthesizing them molecules is often resource intensive requiring several years of time and costing several million or even hundreds of millions of dollars. Hence, the pharmaceutical industry leverages the power of artificial intelligence (AI) to come up with models that can predict the activity of a candidate towards a molecular target. [1]

Several AI approaches such as random forests,[2] SVM[3] and Artificial Neural Networks[1] have been implemented in the past to address this problem. Though several of these approaches have shown promising results in the past, one of the important drawbacks of these methods is their prediction accuracy and time of running. Due to the inherent complexity of a biological system, even a modest R^2 value of around 0.3 in some cases is good enough.[4] Moreover, the ever-increasing size of the QSAR datasets consisting several hundred thousand molecules and thousands of descriptors necessitates a more efficient algorithm that can predict the molecular activity using fewer descriptors in lesser time. In this current work, I propose a feature selection algorithm based on genetic selection followed by hyperparameter tuning with the reduced features as inputs. The proposed method provides an approximate 5% increase (in the tested set) in prediction accuracy compared to the previous model that won the Kaggle competition when the dataset was first released in 2012. Since the activity prediction is a regression problem, I use a deep convolutional neural net (DNN) based approach using the keras module (with TensorFlow backend). The original dataset had 15 different targets. However, due to time and memory constraints, I've selected a subset of 5 datasets from these 15 for the study.

3. Dataset Description and Models used

https://www.kaggle.com/c/MerckActivity

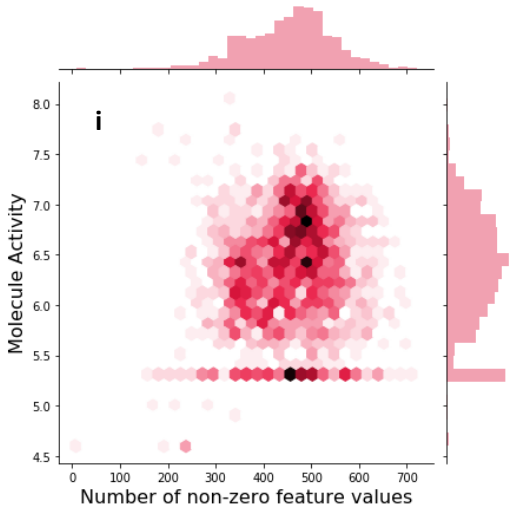
The data that I used in the current project is from the Merck Molecular Activity Challenge, first released on Kaggle in 2012. The objective of this competition was to identify the best statistical model that can predict the molecular activity of potential drug candidates. The dataset originally consisted of 15 molecular activity sub-datasets. The first column is the molecular ID, the second column is a molecular activity (the **y** value). Column 3 to end are the molecular descriptors. In the current project, I am considering 5 out of the 15 datasets originally used. **Table 1** below briefly describes the data with fields that include the name (ID), type (either a drug target or pharmacokinetics (ADME) was considered), description of the dataset, no. of potential drug candidates in each of the datasets and the number of descriptors in each. Due to the confidential nature of the dataset, the identity of the molecules or the features are not known. However, the features for QSAR studies are designed to capture the topological, geometrical, thermodynamic, chemical and electric properties of a molecule.

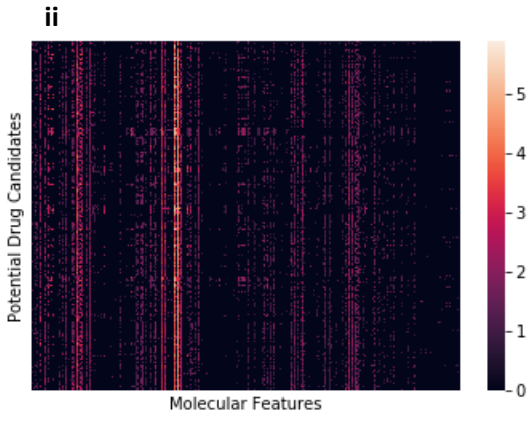
Dataset ID	Type	Description	Number of Molecules	Number of unique descriptors
DPP4	Target	Inhibition of dipeptidyl peptidase 4	8327	5203
HIVINT	Target	Inhibition of HIV integrase in cell assay	2421	4306
HIVPROT	Target	Inhibition of HIV protease	4311	6274
METAB	ADME	Percent remaining after 30 min microsomal incubation	2092	4595
OX1	Target	Inhibition of Orexin 1 receptor	7135	4370

3.1 Data Preprocessing:

- **Features**: Tried two approaches-no feature selection and feature selection based on genetic algorithms. Also, only features that are present in both train and test data are considered to remain consistent.
- **Descriptors**: A $y = log_2(x+1)$ transformation was performed on the descriptors. A binary transformation of y=1 if x>0, 0 otherwise was tried too. However, this gave suboptimal results.

Few observations from the preliminary data analysis:





- From the **fig i** above, it is apparent that there is no discernible correlation between the Molecular Activity and number of non-zero feature values in a molecule.
- Secondly, from **fig ii**, it seems that most of the features are null values and there exists a strong correlation between several features.

From a statistical model point of view, it makes sense to eliminate features that either contribute nothing or contribute in a similar fashion to an existing feature. Redundancy in features often leads to overfitting the data and increases the run time significantly. Hence, I performed a feature reduction followed by hyperparameter optimization that leads to a more efficient and accurate model.

3.2 Model: Deep Neural Network based network was used in this study. The same model as used in the competition and the publication by Ma et al. was used. Briefly, a vector of features represented by $X = [x_1, ...x_N]^T$ for each molecule is used as the input for the neurons of the deep learning model. The output corresponding to the individual molecule under consideration is represented as follows:

$$O = f(\sum_{i=1}^{N} w_i x_i + b)$$

The following parameters were used to train the deep learning model:

- **Network Architecture**: 4 hidden layers. Number of neurons per layer: 4000, 2000, 1000 and 1000 respectively. Activation Function-ReLU
- **DNN training**: **Epochs**: 25,50,75; **Mini-Batch**: 100, 200 and 300.
- **Optimizers**: **Adam** and **rmsprop**

3.3 Genetic Algorithm: Genetic Algorithms (GA) perform feature selection analogous to natural selection in biology. Almost all genetic algorithms have the following components:[5]

1. Fitness function for optimization
2. Population of genes
3. Selection of fit genes for reproduction
4. Crossover between chromosomes (collection of genes) to produce next generation of offspring
5. Random mutation among chromosomes to produce newer chromosomes in the next generation.

Fitness function is the specific function being optimized to identify the best combination of features. In the current study, I used the function `sklearn.svm.SVR` from the scikit learn package for optimization. The resultant features were selected based on their accuracy in predicting molecular activity. The term *gene* represents the different combinations of features (usually encoded in binary) that correspond to the candidate solution. Say if a dataset has N parameters (N_p), then the genes are encoded in the following way:

$$Gene_i = [p_i^1, p_i^2, ...p_i^N]$$

Where p_i^N represents the value of the N^{th} parameter. Usually, this is encoded as either 0 (the feature is not considered) or 1 (the feature is considered). A specific set of genes together constitute a chromosome (4 in this case). The first round of iterations consists of chromosomes with randomly selected genes. The selection of fit genes from a chromosome occurs based on how good the selected genes of a chromosome contribute towards the fitness of an organism (the fitness function in this case). During the optimization process, a crossover between the chromosomes that produce a better model (fitter offspring in this case) are selected. This crossover between

different parent chromosomes that give a fit offspring leads to an enrichment of genes (the individual features) that lead to an overall better model. This method of feature selection is analogous to the Darwinian model of natural selection. Mathematically, the probability that a chromosome gets selected is represented by:

$$P(C_i) = \left| \frac{f(C_i)}{\sum_{i=1}^{N} f(C_i)} \right|$$

Where $f(C_i)$ is the value of the fitness function for the i^{th} chromosome.

Finally, the random mutations among the chromosomes ensure that the fitness function does not converge too quickly and ensures that important features (genes) do not get left out.

3.4 Evaluation Metric:

The performance of all the models was evaluated using the correlation coefficient (R^2) given by the following formula. Here ***x*** is the known activity, $\bar{x}$ is the mean of the known activity, ***y*** is the predicted activity and $\bar{y}$ is the mean predicted activity. N_s is the total number of molecules in the dataset. The model with the best-optimized parameters will be evaluated based on this R^2 metric.

$$R^2 = \frac{1}{15}\sum_{s=1}^{s} r_s^2 \qquad r_s^2 = \frac{[\sum_{i=1}^{N_s}(x_i - \bar{x})(y_i - \bar{y})]^2}{\sum_{i=1}^{N_s}(x_i - \bar{x})^2 \sum_{i=1}^{N_s}(y_i - \bar{y})^2}$$

The entire workflow for analysis is provided in **Appendix I**

4. Data Analysis and Results

All the analysis is performed in python using TensorFlow with **keras**. **Table 2** shows the R^2 and total cumulative time taken to run the models for both the original approach and the one using feature reduction by GA. In the first part of the evaluation, the dataset with reduced features was run with the exact same model parameters as the original approach. As indicated in the table (and **fig.4**), this resulted in approximated 1.6% improvement in the R^2 value. However, this may not be the most ideal approach. As the model performance could be highly variable based on the input features. Since the GA approach changes the input feature vector, a hyperparameter tuning was performed on one of the datasets (METAB) to identify the best combination of parameters to optimize the DNN model. Upon trying several different combinations, the one with **Epoch**: 75, **Activation**: Adam and **Mini Batch** : 100 turned out to be the best combination with an R^2 value of **0.9717** (**fig 3**).

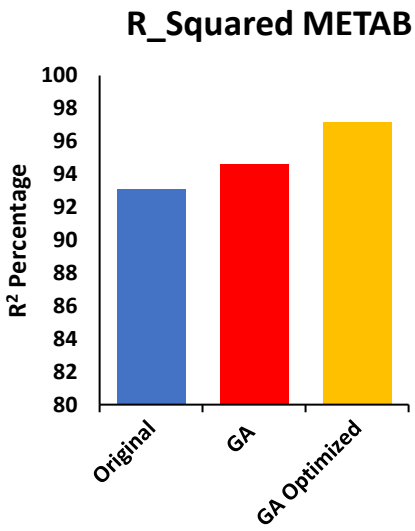


Figure 3. *Results from the hyperparameter optimization.*

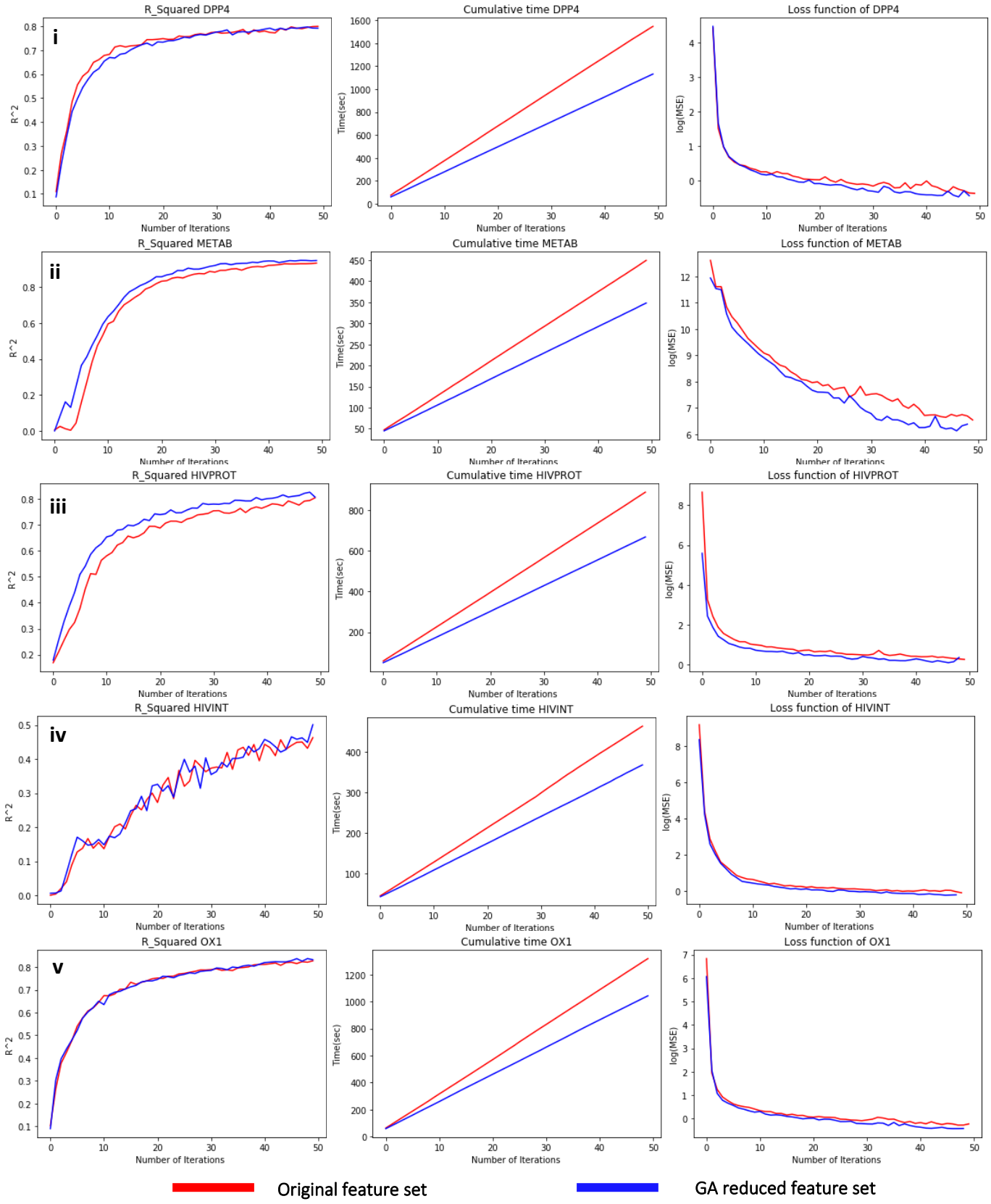


Figure 4. *Comparison of the R^2, cumulative computational time (seconds) and log(mse) (loss function) for all the five datasets analyzed.*

Dataset ID	R^2	R^2 GA	Cumulative time (sec)	Cumulative time (GA) (sec)
DPP4	0.799	0.791	1547.67	1131.45
HIVINT	0.462	0.501	462.35	367.24
HIVPROT	0.804	0.806	887.36	667.38
METAB	0.931	0.946	449.207	347.996
OX1	0.827	0.832	1317.634	1042.112
Average	**0.764**	**0.775**	**932.844**	**711.235**

Table2: Table with the individual R^2 values and cumulative runtime for all the 5 models tested. An average of all the R^2 values and cumulative time taken in also reported.

Also, from the above table, even without hyperparameter tuning, 3 out of 5 models (HIVINT, METAB and OX1) with GA based feature reduction significantly outperformed the original model.

Secondly, due to the feature reduction, there is a 25% percentage improvement in run time on average for all the models. Though the feature selection process takes a similar amount of time as running the actual model, this selection must be done only once. During the hyperparameter optimization, the faster model fitting quickly adds up while trying hundreds of different combinations for parameter optimization. This leads to an overall better time efficiency in the reduced feature set and simultaneously leads to better model accuracy.

5. Discussion and Conclusion

In the present work, I Implemented a new feature selection algorithm based on genetic selection and compared the performance with the existing published method. For the QSAR dataset chosen in the current work, a significant portion of the features are correlated with each other, constants (and/or null values) or have very little variance. Such features usually do not contribute much towards being able to predict whether a molecule is active or not. Having them in the input data could lead to overfitting and increase the overall runtime. In the current study, I showed that feature reduction using GA approach not only reduces the overall runtime but also improves the prediction accuracy. The GA feature selection approach has resulted in a slight (1.65%) improvement over the previously established approach when evaluated with the same model parameters. However, upon tuning the hyperparameters (number of epochs, number of batches and different optimizers) on the reduced feature dataset, a much more significant improvement (~5%) was observed. Though I only used a subset of the total dataset, this approach can be successfully scaled up to produce similar results on the overall dataset. Hence feature reduction by genetic selection-based approach can potentially result in better models that can predict the drug activity with a much greater accuracy. In future, it would be interesting to compare this with other feature reduction approaches such as principal component analysis (PCA) to compare the model effeciencies.

6. References

(1) Chen, H.; Engkvist, O.; Wang, Y.; Olivecrona, M.; Blaschke, T. The Rise of Deep Learning in Drug Discovery. *Drug Discov. Today* **2018**, *23* (6), 1241–1250. https://doi.org/10.1016/J.DRUDIS.2018.01.039.

(2) Vladimir Svetnik, *,†; Andy Liaw, †; Christopher Tong, †; J. Christopher Culberson, ‡; Robert P. Sheridan, § and; Feuston‡, B. P. Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. **2003**. https://doi.org/10.1021/CI034160G.

(3) Wen, L.; Li, Q.; Li, W.; Cai, Q.; Cai, Y.-M. A QSAR Study Based on SVM for the Compound of Hydroxyl Benzoic Esters. *Bioinorg. Chem. Appl.* **2017**, *2017*, 1–10. https://doi.org/10.1155/2017/4914272.

(4) Ma, J.; Sheridan, R. P.; Liaw, A.; Dahl, G. E.; Svetnik, V. Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships. *J. Chem. Inf. Model.* **2015**, *55* (2), 263–274. https://doi.org/10.1021/ci500747n.

(5) Carr, J. *An Introduction to Genetic Algorithms*; 2014.

Acknowledgment: The python code was adapted from the one by George Dahl's team who won the original Kaggle competition. Genetic algorithm is adapted from the GitHub python module by Ahmed Gad. All other modules used were from keras, TensorFlow and scikitlearn.

Appendix I workflow

A) Custom loss function

Define a custom R_squared metric for the DNN model to measure the correlation.

B) Running the model

for *each of the datasets in DataSets[1-5]* ***do***

1) Pre processing

Input: *Original feature vector with N features*

- *Keep the M (≤N) features that are common to both train and test set.*
- *Perform a log(x+1) transformation on all the y (molecular activity) values.*

Output*: New feature vector with M features and log(x+1) transformed activity values*

2) Genetic Algorithm

Input: *Feature vector with M features*

while *(!stopCondition)* ***do***

- *Randomly select a population of genes*
- *Selection of fit genes for reproduction based on their contribution towards predicting molecular activity*
- *Crossover between chromosomes to identify next generation of offspring*
- *Random mutations to produce new set of chromosomes*

Output: *Reduced feature set with L (≤M) features that are fitter than the rest*

3) Deep Neural Network

Input: *Feature vector with M features, log transformed molecular activity values, number of neurons, activation function, optimizer, loss (MSE) and evaluation metric (R^2)*

- *Every feature vector is randomly divided into two parts. 0.8 is used for training and 0.2 for validation.*
- *Evaluate the prediction accuracy using the defined metric for both feature sets consisting* ***M*** *and* ***L*** *features (from GA)*

for *params in GridSearchParams* ***do***

- *Evaluate the performance for each* ***params***
- *This is done only to the GA reduced feature set*

Output: *Value of loss function (MSE) and* R^2 *and the corresponding hyperparameters for all the tested values*