Jagiellonian University

Doctoral Thesis

Development of universal data representations with application in chemistry

Author:
Magdalena Wiercioch

Supervisor:
Jacek Tabor Ph.D. Prof.

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the discipline of science:
Information and Communication Technology

Kraków, 2023

Declaration of Authorship (in Polish)

Ja niżej podpisana, Magdalena WIERCIOCH, oświadczam, że przedłożona przeze mnie rozprawa doktorska pt. "Development of universal data representations with application in chemistry" jest oryginalna i przedstawia wyniki badań wykonanych przeze mnie osobiście, pod kierunkiem prof. dr. hab. Jacka Tabora. Pracę napisałam samodzielnie.

Oświadczam, że moja rozprawa doktorska została opracowana zgodnie z Ustawą o prawie autorskim i prawach pokrewnych z dnia 4 lutego 1994 r. (Dziennik Ustaw 1994 nr 24 poz. 83 wraz z późniejszymi zmianami). Jestem świadoma, że niezgodność niniejszego oświadczenia z prawdą ujawniona w dowolnym czasie, niezależnie od skutków prawnych wynikających z ww. ustawy, może spowodować unieważnienie stopnia nabytego na podstawie tej rozprawy.

Miejscowość i data:

____

Podpis:

____

"When you remove the fear of failure, impossible things suddenly become possible. If you want to know how, ask yourself this question: 'What would you attempt to do if you knew you could not fail?...' "

Regina Dugan

JAGIELLONIAN UNIVERSITY

Abstract

Information and Communication Technology

Doctor of Philosophy

Development of universal data representations with application in chemistry

by Magdalena WIERCIOCH

In recent years, deep learning models have shown their great potential in the field of representation learning. Unfortunately, unlike the established deep learning-based methodologies that have achieved human-level accuracy in various application domains such as computer vision and speech recognition, the development of molecular modeling is still at an early stage. This seems to be mainly caused by the inductive biases of molecules that are completely different from those of image, and the lack of sufficiently large and reliable chemical data.

In this Thesis, we propose solutions for three different tasks:

- *The classification task.* Here, a new model, called HybNN, is applied to detect bioactive chemical compounds.
- *The classification, regression, and interpretability task.* Here, a new model, called SENN, is applied to predict molecular toxicity.

- *Deep representation learning of graphs and sequences*. Here, a new model called TENN is applied to evaluate whether the candidate drug and a target protein are interacting.

First, we propose an architecture that aims to solve the classification problem as its main task. To illustrate how the model works, we apply it to detect bioactive chemical compounds. In contrast to the state-of-the-art methodologies, our approach automatically learns a mixed molecular representation from both physiochemical properties and contextual information that describes the structure of the molecules. In more detail, the contributions of this work are as follows.

- The first thorough comparison of classification approaches with application to molecular bioactivity prediction task.
- A collection of molecular features that aim to capture the structure-property relationships.
- Results demonstrating that a unified approach that employs the bidirectional gated recurrent component and a spatial graph component improves the state-of-the-art for the classification task.

Second, we study both a classification and a regression task. In addition, we focus on the model's interpretability. In the context of chemical data, a typical regression problem is molecular properties prediction. Specifically, we discuss the exemplary application, i.e., toxicity prediction is discussed. In this contribution, we employ a subgraph embedding component that enables one to exploit a graph structure. Also, there was an opportunity to see whether incorporation of a set of global molecular features can be profitable. In particular, we make the following contributions.

- We introduce a novel well-designed deep learning-based architecture that uses a subgraph embedding component that is fed into graph convolutional networks to improve the learning process.

- We study and compare the characteristics of eight different model architectures (RF, SVM, FFN, GIN, GCNN, TopTox, Weave, and SENN - our methodology) employed to solve both classification and regression problems.
- We compile a comprehensive list of molecular features allowing us to improve discriminative capability of our method.
- Finally, we demonstrate the utility of our methodology and find that we achieve remarkably superior performance over the state-of-the-art models on various publicly available regression and classification benchmarks.

At last, deep representation learning of graphs and sequences is our challenge. The proposed deep learning-based approach is evaluated on representative chemical datasets to build a classifier capable of predicting drug-target interactions. This is a preliminary work to see if the composition of sequential information, graph-based structure, and a continuous bag of molecular words could enable to detect more complex interactions. In summary, the contributions of this work are as follows.

- We present the first study on the representation learning of graphs and sequences for the classification task and apply it to the drug-target interaction prediction problem (DTI).
- The novel model employs three units to learn the representations of the drug, target, and chemical compound level by level, and then made prediction with overall interaction representation.
- We find that extraction of the global information of protein sequences and drug compounds leads to improvement in the efficiency of DTI,

and enables to detect more complex interactions. Specifically, the experimental results on publicly available datasets demonstrated the competency of our method.

Streszczenie

Opracowanie uniwersalnych metod reprezentacji danych do poprawy jakości metod uczenia maszynowego z przykładowym zastosowaniem w chemii

W ostatnich latach modele bazujące na głębokim uczeniu sieci neuronowych okazały się przydatne w dziedzinie uczenia reprezentacji (ang. *representation learning*). Wyraźny sukces takich algorytmów widoczny jest w obszarze wizji komputerowej czy rozpoznawania głosu, gdzie zaproponowane komputerowe metody osiągnęły wyniki poziomu eksperta. Niemniej jednak okazuje się, że wykorzystanie dostępnych metod do modelowania związków chemicznych nie jest trywialnym zadaniem. Powodów takiego stanu jest kilka, ale należy do nich fakt, iż molekuła jest zupełnie innym obiektem aniżeli na przykład obraz. Ponadto w problemach chemicznych liczba poetykietowanych danych treningowych jest wyraźnie mniejsza (ang. *inductive bias*).

Dlatego *głównym zadaniem badawczym podjętym w niniejszej pracy doktorskiej jest zaproponowanie metod uczenia reprezentacji, które mają być przydatne m.in. w dziedzinie projektowania leków*. Wychodząc naprzeciw postawionym wyzwaniom Autorka pracy przedstawiła trzy algorytmy pozwalające na uzyskanie komputerowej reprezentacji molekuły. W celu oceny przydatności modeli, wykonano szereg eksperymentów obejmujących zadania klasyfikacji oraz regresji.

Uściślając, w pracy zaproponowano rozwiązania trzech następujących zagadnień informatycznych.

- *Zadanie klasyfikacji.* W tym obszarze nowy model o nazwie HybNN użyto do wykrywania związków chemicznych aktywnych biologicznie.
- *Zadanie klasyfikacji, regresji and interpretowalności.* W tym obszarze nowy model o nazwie SENN użyto do przewidywania toksyczności związków chemicznych.
- *Uczenie reprezentacji grafów i ciągów bazujący na koncepcji głębokiego uczenia.* W tym obszarze nowy model o nazwie TENN użyto do sprawdzenia czy molekuła i biologiczny cel wchodzą z sobą w interakcję.

Pierwszą propozycją jest architektura rozwiązująca problem klasyfikacji. W celu sprawdzenia przydatności modelu wykonano szereg eksperymentów na danych chemicznych, tj. związkach aktywnych i nieaktywnych biologicznie. W efekcie wprowadzona metoda uczy się reprezentacji w oparciu o atrybuty fizykochemiczne związku chemicznego oraz informacje odnoszące się do struktury molekuły. Wkład zaproponowanego podejścia podsumowany został poniżej.

- Dokonano porównania algorytmów klasyfikacji z zastosowaniem do wykrywania związków aktywnych biologicznie.
- Zaproponowano zestaw cech związanych z molekułą, które mają pozwalać na identyfikację zależności struktura a właściwość.
- Rezultaty wskazują, że zintegrowane podejście łączące sieć rekurencyjną z podejściem grafowym wykorzystującym informacje przestrzenne przynosi poprawę wyników w stosunku do innych znanych metod.

W drugim etapie prac skupiono się na zadaniu klasyfikacji i regresji. Dodatkowo poruszona została tematyka interpretowalności modelu. Tym razem w kontekście chemii podjęty został temat przewidywania właściwości związków chemicznych na przykładzie toksyczności. Zaproponowany algorytm zawiera komponent związany z budową podgrafów, który umożliwia bardziej szczegółową analizę grafowej struktury molekuł. Przy okazji eksperymenty ujawniły, że umieszczenie informacji związanej z globalnymi cechami związków chemicznych również poprawia zdolność predykcji systemu. W skrócie, zaproponowana metodologia wnosi następujący wkład.

- Wprowadzono nową architekturę bazującą na głębokich sieciach neuronowych, która zakłada użycie komponentu operującego na podgrafach oraz globalnych cechach związków chemicznych.
- Zaproponowana metoda została porównana z siedmioma różnymi i często stosowanymi architekturami (RF, SVM, FFN, GIN, GCNN, TopTox, Weave) w zadaniach klasyfikacji oraz regresji.
- Wyodrębniono zestaw cech molekuł do poprawy zdolności dyskryminacyjnych podejścia.
- Na podstawie eksperymentów wykonanych na danych dostępnych publicznie wykazano, że algorytm poprawia rezultaty osiągane przez inne metodologie.

Trzecim podejściem zaprezentowanym w niniejszej dysertacji jest głębokie uczenie reprezentacji na grafach i ciągach. W tym przypadku podjęto się zaprojektowania modelu dedykowanego zadaniom klasyfikacji do zautomatyzowanego przewidywania występowania interakcji lek - cel biologiczny. Metoda pozwala sprawdzić czy zestawienie informacji zawartej w sekwencjach białek, grafowej postaci związku chemicznego i tekstowej reprezentacji molekuły pozwala wykryć złożone interakcje pomiędzy obiektami. Wkład tej części jest następujący.

- Zaproponowano innowacyjną metodę uczenia reprezentacji na grafach i ciągach dla zadania klasyfikacji na przykładzie detekcji występowania interakcji lek - biologiczny cel.

- Zaprezentowane podejście łączy trzy oddzielnie skonstruowane komponenty: reprezentację sekwencji białek, reprezentację związku chemicznego w postaci grafu oraz tekstową reprezentację molekuły.
- Zaprojektowane i wykonane eksperymenty wskazują, że wyodrębnienie globalnej informacji dotyczącej związku chemicznego i białka jako celu biologicznego powoduje wyraźną poprawę wyników klasyfikacji. Co więcej, zastosowane podejście można użyć do wykrywania bardziej złożonych interakcji.

Contents

List of Figures

List of Tables

Part I

MOTIVATION AND BACKGROUND

Chapter 1

Introduction

1.1 Context

Over the years, scientists have noticed that the choice and quality of the data representation, or features in the data used to train a machine learning model directly affect the final performance of the used approach. It is also not surprising that algorithm's usefulness depends on the task. However, one can always indicate sets of features considered as representative that are treated as reflection of what the data is like. Then, these features could be used as input for various tasks such as classification or prediction. Therefore, working on learning representation, in some cases, can be beneficial, for example, when data featurization is employed, especially dealing with small datasets.

In general, the concept of representation learning means learning a parameter-function map from the raw input data domain to a feature vector or tensor. The goal is to detect and extract abstract, or higher conceptual level ideas in order to boost the performance of a system over the unseen data. What is more, the dimensionality of the input domain is usually high since objects such as videos, images, or text are taken into consideration. However, the encoded representation is associated with a low-dimensional manifold. In this regard, although there are many dimension reduction techniques that offer the ability to make high-dimensional data space simpler, such methods often do not capture a mapping that is relevant for new data samples. Interestingly, representation learning is developed for doing this job.

In conventional machine learning, we begin with a specific challenge for which there is training data available. Then, the data is pre-processed, transformed, fed into the machine learning pipeline, and a solution is returned. Here, the learning part includes only making decision based on the approximation of the data unknown mapping. In turn, one of the driving factors of the success of deep learning lies in its ability to learn compact and expressive representations directly from the observed data. Furthermore, the availability of programmable highly-parallel hardware, especially graphics processing units (GPUs), caused that hand-crafted features have been replaced by feature learning mechanism. In consequence, the development of architecture-engineering has had a tremendous influence in the field of representation learning. In addition, in the last few years, a number of novel deep learning architectures and building blocks have been published reporting superior performance.

First of all, there is no single definition of what it means to learn a representation. Undoubtedly, an intuition is that a good representation makes the learning task easier. A few years ago, Bengio, Courville and Vincent [12] focused on a few essential aspects of good representations. According to their investigation, a list of prior factors can be introduced. Examples include local smoothness of input, spatial and temporal coherence in a sequence of inputs is observed, or a hierarchical organization of multiple explanatory factors. In addition, factors are related to each other through simple, usually linear dependencies, and factors that are shared with other tasks, also share the statistical power across tasks.

Following these guidelines, deep learning has introduced a few extra requirements to support the field of representation learning and learn a good representation. Examples of such principles are the following:

- **Abstraction and Invariance**. Because the same input is expected to consistently generate the same output, a good representation is supposed to yield more abstract concepts. In consequence, it should be more invariant to small and local modifications in input data.
- **Distribution**. Representations should be expressive and lead to capturing the diversity input space.
- **Disentanglement**. A disentangled representation keeps the information about the elements in a dataset in a form that is interpretable and compact.

The significance of representations of molecules has attracted a great deal of interest in the decades of drug discovery research [165]. A molecule is commonly-seen as a group of atoms held together by bonds. Unfortunately, this representation is itself insufficient for understanding chemical space and solving various problems such as properties prediction [9]. Therefore, given the role and applications of molecular design, several new approaches have to be explored. As a result, in this Thesis, a molecular representation $\mathcal{R}$ is a mapping from drug-like molecules $\mathcal{M}$ to some set X.

Also, many studies led to the development of a wide variety of notations to describe chemical compounds. For instance, one may give an example of a formula arranged in a standardized order called the Hill System Order. Although it seems simple and is commonly used, it has its serious cons. First of all, it lacks information about the atom's links to other atoms, or atom's chemical behavior. Let's take butane and 2-methyl propane (isobutane) as an example. Both structures have the same molecular formula, i.e. C_4H_{10}, but they have different chemical properties. In consequence, just looking at the molecular formula leads to confusion.

Much effort has been made in cheminformatics to develop hand-crafted features for molecular representations called descriptors [147]. Each existing descriptor aims at reflecting structural similarities and biological activities of chemical compounds. The widely-used Coulomb Matrix [129], Bloom filters [14], Extended-Connectivity Fingerprints [125], or Bag-of-Bonds [66] are just four examples of such representations for this purpose. Furthermore, with the advent of computers, the operations including computational storage, retrieval and searching for structures have come into prominence. In addition, using fixed descriptors, researchers made a strong improvement to prediction of molecular properties and activity, considered as one of the most basic and important drug discovery tasks. Nevertheless, hand-engineered features have disadvantages, like large dimension of the feature vector and considering only a certain task. To be more precise, in order to predict drug lipophilicity, the experts would likely take into consideration other characteristics of chemical compounds than to predict toxicity. Thus, the ideal solution could be a mechanism that itself recognizes the discriminative patterns and creates representation for a specific challenge.

In fact, predicting molecular properties, activities, and interactions from molecular structures is one of the fundamental issues in the cheminformatics community [30]. Predictive models are extensively used to support the initial stage of drug discovery and help researchers to assess safety as well as any side effects for safe dosage ranges. Recently, inspired by the remarkable success of machine learning in many tasks including computer vision or natural language processing, researchers have also shown the potentials of deep learning for drug discovery. In particular, deep learning-based architectures are flexible and capable of handling diverse input data types. In this way, they improve the conventional representations in a variety of tasks.

In spite of the notable advantages of deep learning, challenges in applying deep learning to the cheminformatics domain still remain. For instance, data remains an open challenge. Firstly, the enormous space of valid chemical compounds is estimated to be around 10^{60} [15]. Given the vastness of drug-like chemical space, we need efficient and automated methods for development for various applications. Furthermore, the training data is limited for the current challenges in drug discovery. Another aspect that is relevant and should be noted is data bias and data imbalance problems. In addition, in contrast to computer vision or natural language processing fields, the acquisition of the labels to the specific problem is significantly harder to many orders of magnitude. It is caused by the fact that the labels can only be obtained through lab experiments.

In view of the above, in this Thesis, we elaborate upon representation learning, whereby our focus is on three big topics: classification, regression, and deep representation learning of graphs and sequences. Besides, model interpretability is discussed in an ongoing fashion. Our findings are underpinned through several experiments with a focus on the selected serious challenges that exist in chemistry, such as bioactivity prediction, toxicity prediction, and drug-target interaction prediction.

1.2 Tasks studied during the Thesis

The main research question we focus on is: Does learning representation have an influence on improving drug discovery and development process? Therefore, in this Thesis three different deep learning architectures are offered to create a meaningful embedding given a molecular structure. In order to assess the impact of our models, we test them on classification and regression tasks.

Representations learning

As it was discussed, representation learning means learning a parameter-function map from the raw input data domain to a feature vector or tensor, and aims to extract abstract, or higher conceptual level concepts to accelerate improvement in performance of a computational model over the unseen data.

Formulation

Let us suppose that an observation x belongs to a dataset $\mathcal{X}$. Moreover, a space $\mathcal{H}$ denotes the space of representations h and g is a model, i.e. $g : \mathcal{X} \rightarrow \mathcal{H}$. Then, the representations produced by g are used in a larger system to accomplish some specified tasks.

In the context of the Thesis, our aim is to leverage representation learning techniques to tackle the classification and regression task for molecular data.

The classification task

At the core of classification is identification the category to which a new instance belongs, based on a training set of observations.

Formulation

Let $x \in \mathcal{X}$ be an input point and $y \in \mathcal{Y} = \{0, 1\}$ refers to a binary label associated with the negative and positive classes, respectively. Next, assume that we are given a set of positive and negative instances $\{(x_i, y_i)\}_{i=1}^{n}$ drawn independently from the probability distribution with density $p(x, y)$ that is defined on $\mathcal{X} \times \mathcal{Y}$. Additionally, let $f_{cl} : \mathcal{X} \rightarrow \mathbb{R}$ denote a discriminant function to predict a class label for test input point x as $\hat{y} = sign(f_{cl}(x))$.

In this Thesis, we address the classification task employed for molecular bioactivity prediction and drug-target interaction prediction.

The regression task

In contrast to classification, regression focuses on predicting scalar values instead of categorical.

Formulation

Given a regression task $\mathcal{T}$ with a set of training observations $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} \in [a, b] \in \mathbb{R}$, the model is asked to predict the entire regression function across a value range. Please note that y is a continuous quality score in an interval. In addition, the supervised regression task is to find a hypothesis or model f_{reg}: $\mathcal{X} \rightarrow \mathcal{Y}$ that approximates the true relationship between variables and targets as best as possible.

In this Thesis, we address the regression task employed for molecular properties prediction.

1.3 Contributions of the Thesis

It is well worth starting off by clarifying that this Thesis is the candidate's independent contribution. The presented methodologies have not been published before, especially they are not part of the candidate's prior research discussed in Section 1.4. In addition, it's worth mentioning that the shortened form of the methodology introduced in Chapter 3 was accepted for the ELLIS Machine Learning for Molecule Discovery Workshop 2021. Therefore, the shortened form of Chapter 3 is available on the workshop webpage (`https://moleculediscovery.github.io/workshop2021/`). Also, any graphics that appear in this Thesis are those of the candidate.

In this Thesis, we apply the representation learning framework to tasks related to cheminformatics field. In order to differ active chemical compounds from inactives, we propose an architecture that uses bidirectional gated recurrent units (BiGRUs) and a spatial graph representation unit (Chapter 3). The approach is evaluated on a set of experiments on datasets obtained from the publicly available database. The results reveal that our method improves the state-of-the-art for the classification task.

The promising outcomes motivated us to build a framework that exploits a graph structure in more detail to tackle the challenge of toxicity prediction which is seen as a classification and a regression task (Chapter 4). Our algorithm is demonstrated on a widely used datasets, where we improve upon the state-of-the-art. In case of the classification, we beat the other methodologies including RF [17], SVM [153], GIN [169], GCNN [4], and Weave [88] for four datasets. For the regression task, our approach surpasses GCNN [4], TopTox [166], Weave [88] and feed-forward neural network (FFN)) for two datasets. In addition, we focus on the model's interpretability. It shows our method is able to correctly identify the chemical substructures (toxicophores) that may cause toxicity, such as hydrazones.

Then, we investigate how the deep learning-based methodology can be used for building classifiers capable of predicting drug-target interactions. Here, a heterogeneous network is constructed by integrating sequence embeddings and a graph structure (Chapter 5). Here, the results clearly indicate that extraction of the global information of protein sequences and drug compounds leads to improvement in the efficiency of DTI, and enables to detect more complex interactions. Specifically, the experimental results on publicly available datasets demonstrate the competency of our algorithm.

The implemented models are provided online at: `https://bitbucket.org/mgdlnwrch/`.

1.4 Contributions to other projects and publications

While working on this Thesis, the author has also been actively engaged in other research projects. First of all, the author led a project funded by the National Science Centre (Poland) Grants No. 2016/21/N/ST6/01019. The second project was funded under the Iwanowska Program of the Polish National Agency for Academic Exchange (NAWA) - a decision no. PPN/IWA/2018/1/00094/DEC/1.

In the following, the publications in which the author of this dissertation has also been actively engaged by programming, including designing algorithms, designing experiments, running experiments, doing analysis and writing are enlisted.

- **Magdalena Wiercioch**, Johannes Kirchmair, *Dealing with a data-limited regime: Combining transfer learning and transformer attention mechanism to increase aqueous solubility prediction performance*, in *Artificial Intelligence in the Life Sciences* [39]
- **Magdalena Wiercioch**, Johannes Kirchmair, *Deep Neural Network Approach to Predict Properties of Drugs and Drug-Like Molecules*, in *ML for Molecules Workshop at NeurIPS 2020* [161]
- **Magdalena Wiercioch**, *Exploring the Potential of Spherical Harmonics and PCVM for Compounds Activity Prediction*, in *International Journal of Molecular Sciences* [157]
- **Magdalena Wiercioch**, *On modeling objects using sequence of moment invariants*, in *Computer Information Systems and Industrial Management* [159]
- **Magdalena Wiercioch**, *Feature Selection in Texts*, in *International Conference on Computer Recognition Systems* [158]

- **Magdalena Wiercioch**, *Towards Learning Word Representation*, in *Schedae Informaticae* [160]
- Marek Śmieja, **Magdalena Wiercioch** *Constrained clustering with a complex cluster structure*, in *Advances in Data Analysis and Classification* [138]
- **Magdalena Wiercioch**, Marek Śmieja, *Mixture of metrics optimization for machine learning problems*, in *Schedae Informaticae* [106]
- **Magdalena Wiercioch**, Marek Śmieja, Jacek Tabor *Probability Index of Metric Correspondence as a measure of visualization reliability*, in *Online Proceedings of Wrocław University of Science and Technology* [162]

Furthermore, the candidate has taken an active part in several research events. The oral presentations or poster presentations prepared and presented by the candidate are enlisted.

- *Deep Neural Network Approach to Predict Properties of Drugs and Drug-Like Molecules*, ML for Molecules Workshop at NeurIPS 2020, 2020, Vancouver, Canada
- *Detection and characterization of active compounds based on Random Matrix Theory*, Conference Random Matrix Theory: Applications in the Information Era, 2019, Kraków, Poland
- *Learning Object Descriptors with Application in Cheminformatics*, WiML Workshop at NeurIPS 2018, 2018, Montreal, Canada
- *Automated de-novo molecule design based on Deep Neural Networks*, 14th German Conference on Chemoinformatics, 2018, Mainz, Germany
- *On Modeling Objects Using Sequence of Moment Invariants*, 17th International Conference on Computer Information Systems and Industrial Management Applications, 2018, Olomouc, Czech Republic

- *Text Embeddings Based on Synonyms*, 21st International Conference on Text, Speech and Dialogue, 2018, Brno, Czech Republic
- *Representation Learning: A Case Study*, 58th Cracow School of Theoretical Physics, 2018, Zakopane, Poland
- *Text Embeddings Based on Clusters*, 10th Cracow Cognitive Science Conference, 2018, Kraków, Poland
- *Improving adverse drug reaction detection with models combination*, International Language & Technology Conference, 2017, Poznań, Poland
- *Feature Selection in Texts*, International Conference on Computer Recognition Systems, 2017, Polanica-Zdrój, Poland
- *A new density based clustering algorithm*, Meeting of Polish Special Interest Group on Machine Learning of Computer Science, 2017, Kraków, Poland
- *Towards Learning Word Representation*, Theoretical Foundations of Machine Learning Conference, 2017, Kraków, Poland
- *Probability Index of Metric Correspondence as a measure of visualization reliability*, Meeting of Polish Special Interest Group on Machine Learning of Computer Science, 2015, Gliwice, Poland
- *Mixture of metrics optimization for machine learning problems*, Theoretical Foundations of Machine Learning Conference, 2015, Będlewo, Poland

1.5 Thesis organization

The Thesis addresses three different tasks (representation learning, classification, regression) on different kinds of data related to chemistry. The remainder of this Thesis is organized as follows:

- Chapter 2 introduces the concept of cheminformatics, drug discovery challenges and gives the necessary information connected with existing molecular models.
- Chapter 3 is dedicated to the classification problem. We apply our model to detect bioactive molecules. To specify, a supervised approach is introduced that allows one to extract the features of different aspects of chemical compound and capture the structure-property relationships.
- In Chapter 4, we focus on a regression task and study model's interpretability. Therefore, a molecular properties prediction task is introduced, and our approach that employs a subgraph embedding component and global molecular features is described.
- Chapter 5, explores the topic of deep representation learning of graphs and sequences. To demonstrate our novel architecture, we build a classifier that predicts drug-target interactions.
- Chapter 6 concludes the Thesis with a discussion of limitations and future research avenues.
- Appendix A provides background on representation learning, deep learning architectures, and the notation used throughout the Thesis.

Chapter 2

Background: cheminformatics and learning representations for molecular data

In this chapter we provide the necessary background information related to existing molecular models and challenges faced by cheminformatics which are common requirements for the rest of the Thesis.

2.1 Representation learning and cheminformatics

In the context of operations, the machine learning system is comprised of three components where each component is responsible for a separate task [45]. Those are: representation, goal, and optimization (see Figure 2.1).

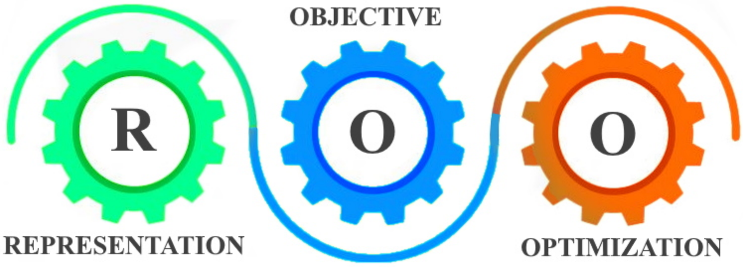


FIGURE 2.1: Three components of Machine Learning Systems.

Thus, the essence of an effective machine learning system is that one has to retrieve valuable information from the input data and somehow transform it into an internal representation such as a feature vector. Also, there is a specified learning algorithm, and a relevant objective function that captures what the system is supposed to do. At the same time, an optimization process should be taken into consideration to get the optimal results.

Obviously, a large body of work has focused on understanding the process of representation learning. It is known that the goal of representation learning is to construct effective data representations for various applications including classification, regression, domain adaptation, among others. The idea is to map high dimensional observations into a lower-dimensional latent space ($\mathbb{R}^d$), such that important properties (like distance) are preserved. For instance, if two objects have similar connections in the original space, their learned vector representations should be close, too.

It shows that deep learning [60] introduces several key innovations from the point of view of representation learning. In particular, two main concepts can be discussed, i.e., distributed representation and deep architecture.

- **Distributed Representation.** Deep learning provides a technique to obtain a low-dimensional real-valued dense vector, called distributed representation. Intuitively, it can be treated as a form of compression, where a large scale input with many features is meaningfully transformed into relatively few dimensions. As such, it enables to represent feature spaces in many different domains in an efficient way.
- **Deep Architecture.** According to the numerous observations, deep neural networks are more powerful than their shallow counterparts on a great variety of machine learning tasks [98]. One inevitable advantage of deep architectures lies in hierarchical learning, in which the layers learn useful representations of the data.

Despite the significant success of deep learning-based methodologies in computer vision, natural language processing, and other domains in data mining and machine learning, cheminformatics poses additional challenges in addressing diverse problems in drug discovery. In fact, drug design is not straightforward and is still in its infancy. Therefore, since the choice of the molecular representation model is considered as a limiting factor of the explainability and performance of the resulting methodology, these challenges led to the development of representation learning research which focuses on applying deep learning techniques to chemical data.

2.2 Cheminformatics and drug discovery

The field of chemoinformatics, also known as chemical informatics or chemoinformatics, provides a set of tools in the computational toolbox that can be applied to drug discovery. In fact, the concept of cheminformatics can be traced back to the 1990s. However, it was firstly defined by Frank Brown in 1998. In his paper entitled *Chemoinformatics: what is it and how does it impact drug discovery* [18], Brown attempts to present the vision of cheminformatics' goals: "The use of information technology and management has become a critical part of the drug discovery process. Cheminformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and organisation.". In fact, cheminformatics has its roots in a few more established fields such as chemistry, chemical information, chemometrics, and computational chemistry. Also, the pharmaceutical industry plays a vitally important role in the development of chemical objects handling. Nevertheless, with the advent of computers, cheminformatics has evolved into a new branch of science that is highly correlated with computer science. With these tools at their disposal, the present focus of cheminformatics is mainly drug discovery.

Drug discovery aims to find novel molecules that bind to a specific protein and affect protein activity (by blocking or enhancing effects), which leads to the modification of the course of the disease. In other words, the key challenge in drug discovery is to identify chemical compounds that satisfy multiple constraints, such as potency, safety, and desired metabolic profiles at the same time. Unfortunately, the process of drug discovery suffers from huge computational cost and time-consuming procedures, which limits its application in the pharmaceutical industry [74, 78, 132]. To give an illustrative example, existing drug discovery pipelines take 5-10 years with a cost counted in billions of dollars. The reasons underlying this delay are poor drug-like properties and complicated molecular properties. In addition, modern drug discovery usually involves a cost-ineffective Virtual Screening (VS) process to select candidates from large chemical databases for further synthesis [102]. Therefore, cheminformatics, especially deep neural network-based techniques, can be a game changer in various areas of CADD (Computer-Aided Drug Design).

Indeed, there are few critical steps in the drug development process, including target identification and screening, lead generation and optimization, preclinical and clinical studies, and registration of a drug (see Figure 2.2). In the early phases of the drug discovery cycle, a foremost step is target validation to prove the selected target is relevant to the disease. This is followed by hit identification approaches to determine a significant set of molecules able to interact with the target. Further, the hit to lead stage delivers compounds with evaluated properties, including affinity to extract lead compounds. After the selection of lead compounds, the lead optimization phase improves the desired properties of molecules (e.g. pharmacodynamics pharmacokinetics). The final stages of drug discovery are called preclinical and clinical development. The first refers to studies occurring prior to clinical testing. For instance, toxicology screenings my be carried out then. Once the pre-clinical research is complete, the newly synthesized drug candidate is moved on to clinical development phase that involves clinical trials to finetune the drug for human use.

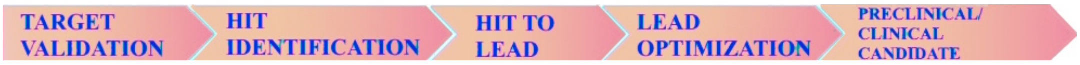


FIGURE 2.2: Key steps of the drug discovery and development process.

The studies in cheminformatics have received increasing attention due to progress in speed and performance. Many machine learning approaches have been successfully applied in a variety of tasks for drug discovery. The examples are molecular properties prediction, drug-drug interaction, or drug-target interaction prediction. One of the fundamental challenges for these studies is how to learn expressive representation from molecular structure [22, 32, 157].

2.3 Molecular representations

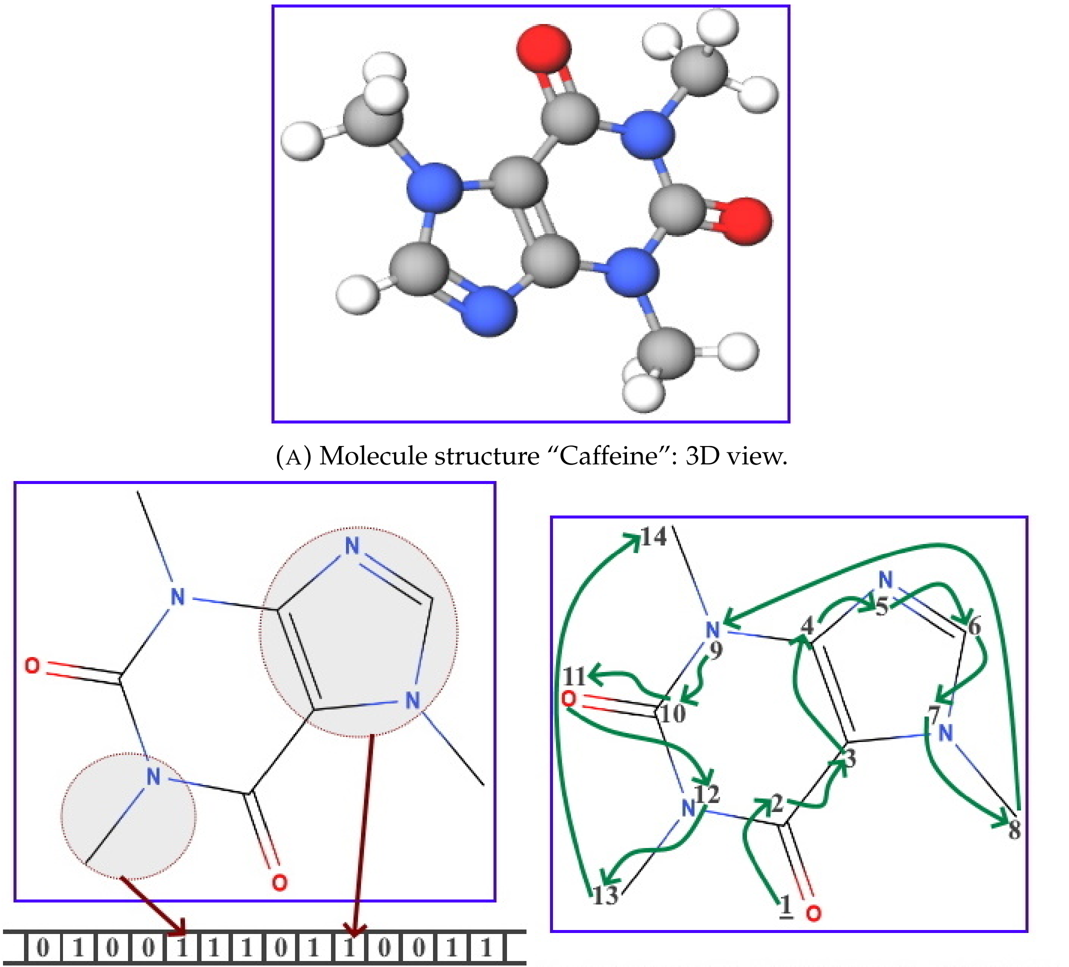
Undoubtedly, a vast number of published models are based on traditional molecular representations [55, 137]. First of all, there is no clear best traditional representation of chemical compounds for machine learning algorithms, and certain representations may be better for specific tasks. Nevertheless, existing traditional molecular representations fall into mainly three categories: SDFs, numerical molecular descriptors, and SMILES strings.

SDFs

The SDF or Structures Data File contains the structural information and associated properties for one or multiple molecules. The advantage of using an SDF is that one can work on 2- or 3-dimensional structures. On the other hand, the disadvantage is that it is a specialised format. In consequence, SDFs are hard to read for people who do not have expertise in the subject domain.

As it was mentioned previously, for decades quantitative structure-activity relationship (QSAR) or quantitative structure-property relationship (QSPR) studies [113, 128] have mainly relied on hand-crafted features for molecular representations to guide the drug discovery process. Their objective is to represent the chemical information of actual molecules in a computer-interpretable vector of numbers before being used to train the machine learning model [146]. In fact, a number of different representations of small molecules exist, and one can debate on the relative pros and cons of the given approach for the purposes of molecular design. For instance, molecular descriptors may be classified into five common categories related to their dimensionality [148].

- **0D** (They are based on molecular formula. Examples: atom counts, bond counts.);
- **1D** (They are based on chemical graph. Examples: fragment counts, functional group counts.);
- **2D** (They extract chemical features from structural topology. Examples: Balaban index, Weiner index.);
- **3D** (They extract chemical features from structural geometry. Examples: GETAWAY, WHIM.);
- **4D** (They take into consideration multiple structural conformations. Examples: Raptor, GRID.);



(A) Molecule structure "Caffeine": 3D view.

(B) Fingerprint representation is a bit vector that consists of 0's and 1's that are associated with the absence or the presence of a particular fragment of the chemical compound structure.

O=C1C2=C(N=CN2C)N(C(=O)N1C)C

(C) SMILES representation. A number is assigned to each atom in the chemical compound and then each atom is visited in the molecular graph in a well-defined order.

FIGURE 2.3: The commonly used molecular representation methods based on the example of Caffeine.

Fingerprints

Descriptors can also be split into fragments known as fingerprints (FPs). Fingerprints represent molecular structure in a vector format, a bit vector that consists of 0's and 1's that are associated with the absence or the presence of a particular fragment of the chemical compound structure. The example of fingerprint representation is depicted in Figure 2.3bB. Obviously, various fingerprints exist to model different aspects of a molecule. Amongst the common types of fingerprints are MACCS and Hashed fingerprints [90]. In general, fingerprints offer a number of advantages, including the fact that they can represent an essentially infinite number of different molecular features, and allow for easier interpretation of analysis results and inference. On the other hand, fingerprints bring difficulty in machine learning methods because of the curse of dimensionality [8] that is related to various problems that arise when one works with high-dimensional data.

Graph representation

One of the most intuitive ways to represent molecular structures are graphs. Therefore, any chemical compound is naturally seen as a mathematical graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ refers to chemical bonds and $\mathcal{V}$ are atoms. A major disadvantage of graph representation is the lack of information related to bond lengths and 3D conformation.

SMILES

The most popular way to represent molecular graphs for machine learning is the so-called simplified molecular-input line-entry system (SMILES) string [156]. Specifically, it is seen as 'chemical language' that encodes structural information of a molecule into single ASCII strings of 20-90 characters (Figure 2.3). A so-called SMILES representation is constructed by assigning a number to each atom in the chemical compounds and then visiting each atom in the molecular graph in a well-defined order. For instance, RDKit [97] employs depth-first search as the graph traversal algorithm. One benefit of SMILES strings is that they encode an exact structural representation of a chemical compound. However, one downside is that SMILES are not injective over chemical compounds. In other words, if two SMILES strings are not the same, it does not imply the underlying molecules are not equivalent. On the other hand, a widely used in the field of cheminformatics, the canonical

SMILES ensures a one to one correspondence between chemical compound and SMILES string. However, some researchers claim that the latent representations obtained from canonical SMILES can be less useful since they may be more associated with specific grammar rules of canonical SMILES rather than with the chemical structure of the given molecule [13]. As a consequence, this may affect interpretation and optimization tasks because it is natural to expect that a latent space represents chemical properties and captures notions of chemical similarity rather than grammar rules.

Then, these representations, after pre-processing, can be used as input to machine learning models such as Support Vector Machines [33], Random Forests [17], Kernel Ridge Regression [112] or neural networks [175]. The next step is building a computational model that reveals the relationship between the chemical compound and a given property or activity. Despite the fact that these approaches have achieved promising results in many tasks including biological activity detection [25] and toxicity risk evaluation [1], encoding chemistry is still the biggest challenge. Here, we give a few examples of the reasons.

1. Our skills to boil down chemistry into simple numbers are very limited.
2. Manually creating relevant features is really time-consuming.
3. Feature engineering is not easy in the case of not well-investigated structure-property relationships [150].
4. Hand-crafted features lack generalizability and scalability since one can make assumptions associated with certain properties of molecules that may be less relevant [52].
5. The more complex descriptors often require high computational cost [16, 47, 159].

Therefore, although research efforts have been put on designing novel handcrafted molecular descriptors during the past years, deep learning can take an important step in the field of representation learning in the context of cheminformatics.

2.4 Upsides and downsides of learning data-driven molecular descriptors for chemoinformatics-related tasks

Given the recent progress in machine—especially deep—learning on numerous tasks [35, 73, 98], it is no wonder that drug discovery could benefit from this. In particular, molecular representation learning [60] seen as the process of automating the discovery of feature representation of molecular structure has attracted significant attention from both chemists and machine learning scientists [27, 172]. Indeed, deep learning is a promising tool to facilitate a variety of downstream applications, including bio-property prediction and chemical reaction prediction, etc. [87, 170, 81]. The driving force behind this lies in the fact that deep learning-based approaches enable us to learn compact and expressive representations directly from the observed data. Generally, current works along the line of deep learning for molecules can be categorized into two main groups according to the input data type of chemical compound,

- string-based methodologies
- and graph-based methodologies.

Specifically, SMILES (simplified molecular-input line-entry system) is a sequence notation encoding a molecule into a character string that follows a specified grammar [156]. As discussed previously in 2.3, although several advantages of using SMILES exist, this format is considered as non-unique [102] since a single molecule can have multiple possible SMILES strings. Nevertheless, a number of deep learning-based architectures work with SMILES since a lot of existing machine learning frameworks have available many techniques for working with text.

However, a chemical compound can also be naturally seen as a graph with nodes corresponding to atoms and edges corresponding to bonds, and one may learn on a molecular structure. Viewing molecule structure as graph data leads to graph neural networks-based (GNNs) [61, 131] architectures. The molecular graph enables us to capture the spatial connectivity of different atoms and preserve molecular bonds which, amongst others, bring real benefits. In consequence, large number of graph-based neural network models have been investigated, such as message passing neural networks (MPNNs) [57], convolutional networks [58], and graph attention networks (GATs) [154]. Obviously, they were successfully employed to tackle learning molecular representation task [65, 167].

Even though a graph-based approach has already led to the development of state-of-the-art improvements, more computational methods are required to handle the chemical structure that could support more effective DNN-based drug discovery. For instance, the scarcity of labeled data brings serious challenges for deep learning in molecular representation. It is caused by errors and the fact that lab experiments are costly [36, 99]. In consequence, training datasets used in cheminformatics problems are often limited in size. In turn, this results in overfitting and finally the learned representations lack of generalizability [75]. For this reason, to address the above issue, one has to design a more powerful models that exhibit scalability and accuracy to express a great variety of molecules. In addition, another problem that needs to be discussed is the limited structural information incorporated into existing deep models. Although treating a chemical compound as a set of atoms and bonds is reasonable, one should take into consideration the fact that it also consists of various molecular dependencies that cannot be missed. In particular, structural dependencies between nodes and edges and interactions must be identified.

2.5 Graph-structured data

The core input data structure considered throughout this work is the graph since a molecule can be represented as a graph. Therefore, here we discuss some basic elements of graph theory, as well as the key concepts required to understand how GNNs are formulated and operate.

Let us consider the graph-structured data as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges and $\mathcal{V}$ is a set of vertices. Here, v_i is the ith vertex and e_{ij} is the edge from the jth vertex to the ith vertex. $N(v_i) = \{u_i \in \mathcal{V} | (v_i, u_i) \in \mathcal{E}\}$ denotes the neighborhood of the ith vertex. The adjacency matrix A is a $n \times n$ matrix, where $a_{ij} = 0$ if $e_{ij} \notin \mathcal{E}$ and $a_{ij} = 1$ if $e_{ij} \in \mathcal{E}$. In addition, when the graph-structured data is employed, one may use a vertex feature matrix X^{vertex} of $n \times f$ scales, and an edge feature matrix X^{edge} of $m \times c$, representing the feature vector of a vertex and an edge, respectively.

A spatial-temporal graph is an attributed graph where the attributes of vertices change dynamically over time. It can be represented as $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}, \mathcal{X}^{(t)})$. In case of the directed graph, it has an asymmetric adjacency matrix since the edges are directed from one vertex to another. In turn, for the undirected graph, all edges are undirected. In consequence, the adjacency matrix is symmetric, and its normalized Laplacian matrix is defined as:

$$L = I_n - D^{-0.5} A D^{-0.5}, \tag{2.1}$$

where D refers to the diagonal matrix of vertex degrees with $D_{ii} = \sum_{j=1}^{n} A_{ij}$, and I_n is an identity matrix. In fact, the normalized graph Laplacian matrix is real symmetric positive semi-definite. Hence, it can be factored as

$$L = UQU^T, \tag{2.2}$$

where $U \in \mathbb{R}^{n \times n}$ refers to the corresponding eigenvectors ordered by eigenvalues λ, and Q denotes the diagonal matrix with $Q_{ii} = \lambda_i$.

Part II

LEARNING REPRESENTATIONS AND APPLICATIONS

Chapter 3

Learning Hybrid Representation for Classification

In this contribution, we propose a novel architecture, namely Hybrid Deep Neural Network (HybNN), based on structural features and physiochemical properties. In addition, a new spatial graph representation unit that aims at processing the spatial graph matrix was put forward. The features were extracted both at the atom level and molecule level, which ensure that both the fine-grained and coarse-grained connectivity information of chemical compounds is provided. Moreover, BiGRUs were adopted to cover more elaborate information. The experiments on classification task on fourteen prevalent datasets support the generalizability and robustness of HybNN, which outperformed state-of-the-art algorithms, including ChemixNet [118], RF [17], SMILES2vec [59], and Chemception [58] for molecular bioactivity prediction.

3.1 Introduction

An important step in the drug discovery pipeline is to predict molecular activity. This is caused by the fact that the discovery of a new drug involves testing small molecules for their ability to bind to the target receptor [20]. Since the task is to separate the active chemical compounds from the inactives, the classification task is usually suggested. As a result, in the last decades, many computational methodologies have been proposed and widely developed to expedite the process of identification of active molecules. In fact, a series of approaches exploring quantitative structure-activity relationships (QSAR) have been developed [40]. Most of them focus on similarity searching - based methods [21]. In addition, typically the development of a reliable computational methodology needs high-quality descriptors [29].

What is more, the introduction of molecular descriptors has opened up new avenues for machine learning - based bioactivity prediction. Notably, Bayesian classifiers [114], feed-forward neural networks (FNNs) [7], Random Forests (RF) [141], and Support Vector Machines (SVMs) [86] are only a few examples of traditional machine learning methods that have made an impact in drug discovery. Despite numerous records of successful application of these approaches in cheminformatics, they come with several limitations. Firstly, the well-established approaches often use a number of irrelevant descriptors and are, in general, not robust for high dimensional data. Another challenge in molecular activity prediction is predictive accuracy that is not good enough to avoid unwanted errors which leads to the inability of the drug to meet the condition for which it was developed.

Albeit powerful, traditional machine learning methods often lead to insufficient outcomes. Recently, there has been deep learning architectures successfully applied on molecular data, too. Nevertheless, in practice, dealing with neural networks poses several unique challenges. First, the available models usually need large and high-quality data. Second, increasing depth and width of deep architectures also has an influence on growth in computation. Third, as the model is not completely aware of the structural information related to the molecule, it cannot infer any significant molecular dependencies. In this chapter, the above-mentioned gap is bridged by providing a workflow procedure, named **H**ybrid **D**eep **N**eural **N**etwork (HybNN). HybNN has the following advantages.

1. HybNN outperforms the other approaches on all test datasets, i.e. fourteen publicly available benchmark datasets. We demonstrate extensive comparisons with various approaches, including traditional machine learning methods and deep learning-based models. Thus, the algorithm sets a new standard in the classification task.

2. HybNN consists of two separate blocks. To specify, the bidirectional gated recurrent units (BiGRUs) and a spatial graph representation unit are combined to extract the features of different aspects of the chemical compound. Thus, it potentially better captures the differences between chemical compounds of similar structure.

3. HybNN uses a collection of molecular features to construct the molecular graph representation of the input. It ensures that the most common attributes will be taken into consideration and better reflect the molecular functionalities.

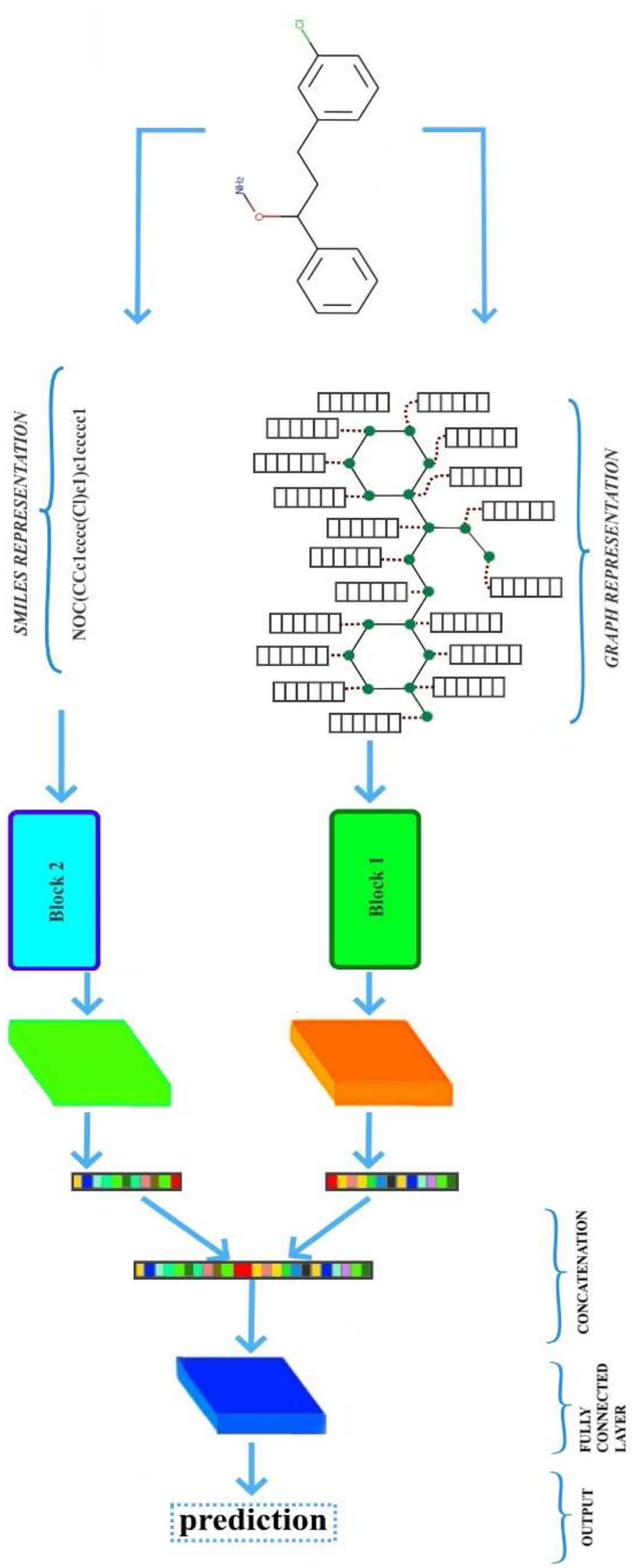


FIGURE 3.1: Schematic of the architecture of HybNN. Our model is comprised of five main modules: the graph embedding module, the word embedding module, spatial graph embedding module (Block 1, see Figure 3.2), the BiGRU network (Block 2, see Figure 3.3), the gathering linear layer.

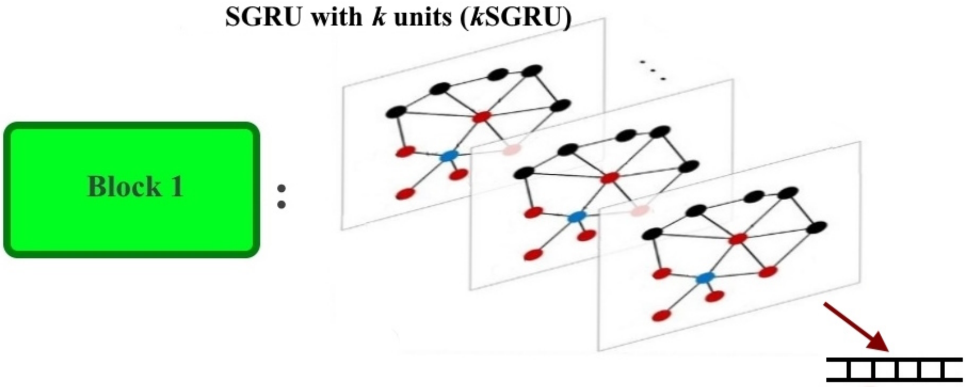


FIGURE 3.2: HybNN: block 1. The unit is based on the idea of graph convolution (see Figure 3.4). Firstly, we construct a vertices-related matrix to preserve the topology structure of the graph $\mathcal{G}$ in the space. Then, to learn hidden representations from the preprocessed graph representation of chemical compounds, an embedding layer is incorporated that provides spatial topology with translational invariance.

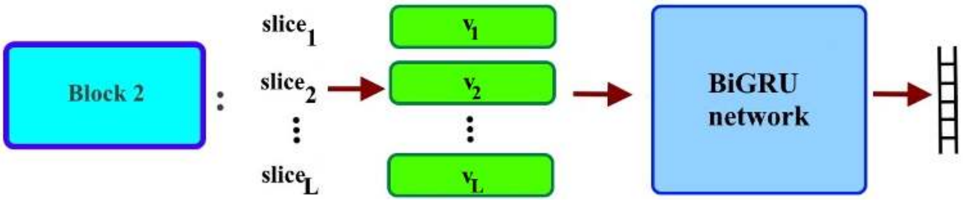


FIGURE 3.3: HybNN: block 2. We create SMILES substrings embeddings. Then, the module takes the sequence of embedded vectors as input and returns a molecular representation.

3.2 Related Work

One of the most popular traditional machine learning methods that is often used to deal with chemistry-related challenges is Random Forests. The idea is to build multiple decision trees by randomly extracting various features and different samples, namely, multiple weak classifiers. In the end, the results of multiple weak classifiers are voted on to get the final outcome. Recently, deep neural network-based architectures have been proposed and made tremendous progress in the drug discovery field due to, among others, their flexibility [104]. The first well-known success was the winning Merck Molecular Activity Kaggle Challenge with a multi-task deep neural network

(MT-DNN) [34]. Dahl et al. constructed a model that trains a single neural network with multiple output neurons, where each of these neurons predicts the activity of the input chemical compound in a different assay. On 14 of the 19 assays, their model obtained an AUC score exceeding the best baseline. Obviously, that work was followed by many other works [48, 57] that demonstrated the power of deep architectures in molecular modeling. In particular, Chemception takes image data from 2D drawings of molecules as input before predicting an output such as the molecular activity [58]. The methodology was inspired by Google's Inception-ResNet for image classification [142]. In fact, the general accuracy of this method across three tasks appears to be comparable with deep neural networks trained on engineered descriptors, including ECFP fingerprints. Next, SMILES2Vec has contributed to existing state-of-the-art learning directly from chemical text representations [59]. In general, the idea is to treat SMILES strings as text sequences and train recurrent neural network architectures. Based on the results, Goh et al. conclude that their SMILES-based algorithm is effective in predicting a wide range of properties since it outperforms the current state-of-the-art in regression tasks and matches in classification tasks. Finally, in 2018, ChemixNet was proposed, where a set of neural networks is used to predict chemical properties by leveraging both SMILES and molecular descriptors (MACCS fingerprints) as inputs [118]. Paul et al. showed ChemixNet is an efficient way to improve model performance and pointed out the efficacy of using mixed input architectures for molecular learning tasks.

3.3 Methodology

In this chapter, we propose a Hybrid Deep Neural Network (HybNN) algorithm which is a novel BiGRU and graph - based neural network applied to molecular structures to learn and predict bioactivity. In this architecture, five key components are included: the word embedding module, the graph embedding module, the BiGRU network, the spatial graph embedding module, and the gathering linear layer. The main architecture is shown in Figure 3.1. The goal of the graph embedding module is to integrate the atom information so as to extract crucial characteristic properties of a chemical compound that help to identify it and capture valuable information from a molecular structure (block 1), see Figure 3.2. In turn, the word embedding module construction allows one to embed the SMILES substrings. The BiGRU network unit then captures local and global SMILES string context information present in the molecular structure (block 2), see Figure 3.3. The preprocessed features are then gathered together. Finally, we employ a fully connected layer and a classification layer to pass the information on whether the compound is bioactive or not.

3.3.1 Graph featurization and CNNs on molecular graph data (block 1)

In dealing with molecular data and graph neural networks, there is no way to explicitly use chemical compounds. There must be a preprocessing step. Hence, we transform all the molecules and convert the input samples to a regular structure. To specify, we define an adjacency matrix A and introduce an initial vertices-related matrix X to preserve the topology structure of graph $\mathcal{G}$ in the space. The matrix $A \in \mathbb{R}^{n \times n}$ corresponds to the connections information between atoms, where n refers to the number of atoms in a chemical compound. Therefore, for two atoms i and j, respectively, $A_{ij} = 1$, if there is a bond (an edge) between i and j, while $A_{ij} = 0$, if not. At the beginning, the values in this adjacency matrix are set to zero. Also, we introduce the matrix $X = \{x_1, x_2, \ldots, x_n\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$, and $x_i \in X$ denotes a m-dimensional vector, encodes m features of n atoms. In the case of

TABLE 3.1: Atom features.

Feature	**Description**
atom type	P, H, O, S, F, Si, C, Cl, Br, Mg, Na, Ca, Fe, As, Al, I, B, V, K, Tl, Yb, Sb, Sn, Ag, Pd, Co, Se, Ti, Zn, N, Li, Ge, Cu, Au, Ni, Cd, In, Mn, Zr, Cr, Pt, Hg, Pb (one-hot)
valence	number of implicit valence
degree	number of directly bonded neighbors
atomic mass	mass of the atom, ranges from 1 to 260
formal charge	integer assigned to an atom in a molecule in the covalent view of bonding
hydrogens	number of hydrogens neighbors
hybridization	sp^2, sp^3, sp^3d
aromaticity	Is the atom a part of an aromatic systems?

an atom type, one-hot encoding comes in handy. In all other cases, the value is assigned to its fixed index position. The atom type, valency, degree, atomic mass, formal charge, or aromaticity are only a few examples of the attributes of each atom that were taken into consideration. They are listed in Table 3.1.

To learn hidden representations h from preprocessed graph representation of chemical compounds, an embedding layer is incorporated that provides spatial topology with translational invariance. Specifically, for each atom, we obtain a spatial graph matrix. To do so, we employ a linear transformation defined as

$$X^F = XW + b, \tag{3.1}$$

where $X^F \in \mathbb{R}^{n \times d}$ denotes the lower-dimensional representations, $X \in \mathbb{R}^{n \times m}$ refers to the high-dimensional initial matrix of attributes formed from the graph representation, $W \in \mathbb{R}^{m \times d}$ denotes the weights matrix, $b \in \mathbb{R}^{n \times d}$ is the bias, m and d are the dimension sizes of X and X^F, respectively. Next, we encode the connection space related to the atoms by defining $X^S \in \mathbb{R}^{n \times n \times d}$:

$$X_{ij}^{S} = \begin{cases} X_j^F & if A_{ij} = 1 \\ 0 & otherwise \end{cases} \tag{3.2}$$

where X_j^F refers to the jth row of the matrix X^F.

In consequence, each vertex is encoded by the matrix $X_i^S \in \mathbb{R}^{n \times d}$ that contains information related to the atom's position and its neighbourhood. Such a construction of the matrix X^S causes that it describes the characteristics of the atoms in the molecule including the spatial arrangement. We demonstrate below SGRUs which take advantage of X^F and X^S.

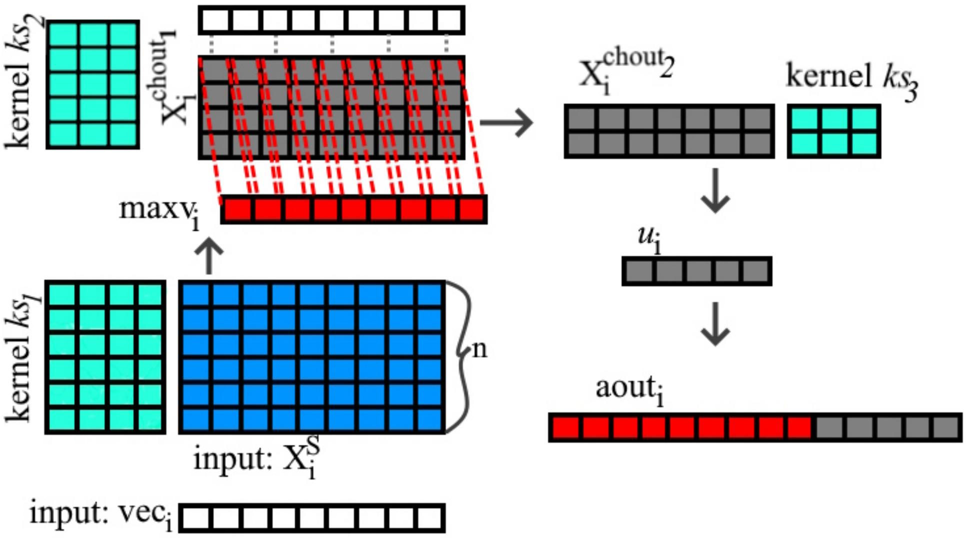


FIGURE 3.4: HybNN: block 1, SGRU. The processing of molecule in SGRU.

Spatial graph representation unit (SGRU)

The spatial graph representation unit (SGRU) is based on the idea of graph convolution. This approach allows one to update the atomic representation, and at the same time, consider both the information connected with a given atom i and ith neighbourhood. Here, the core task is to process the matrix $X_i^S \in \mathbb{R}^{n \times d}$ (that contains spatial information, as mentioned above), and the feature vector $vec_i = X_i^F \in \mathbb{R}^d$ for atom i included in the chemical compound. Specifically, the first step is to set a number $chout_l$ as the number of 1D convolutional output channels, the kernel size ks_l, padding size ps_l, and stride ss_l for each convolutional layer $l \in \{1, \ldots, L\}$. Then, the matrix X_i^S is convoluted to get $X_i^{chout_l} \in \mathbb{R}^{chout_l \times d}$. Here, we distinguish two cases.

- $l = 1$; Firstly, we want to calculate an intermediate atom vector as the output of a function that aims to find a vector $maxv_i = (mv_1, \ldots, mv_d) \in \mathbb{R}^d$ such that $mv_j = \max(X_{i,j}^{chout_l})$, where $X_{i,j}^{chout_l}$ is the jth column vector of matrix $X_i^{chout_l}$ and $j \in \{1, \ldots, d\}$. This operation can be seen as pooling. We further concatenate the vector containing attributes, i.e. vec_i with $X_i^{chout_l}$ and the result, $X_i^{chout_l} \in \mathbb{R}^{(chout_l+1) \times d}$, is fed up to the next $(l+1)$ convolutional layer.
- $l > 1$; The matrix $X_i^{chout_{l-1}}$ is convoluted to get a new matrix $X_i^{chout_l}$ that is, depending on the value of l, fed up to the next (l + 1) convolutional layer. This step is repeated until one obtains a vector $u_i \in \mathbb{R}^{dim}$.

Given the vector u_i and the vector $maxv_i$, these two vectors are concatenated, and the updated atomic vector representation $aout_i \in \mathbb{R}^{dim+d}$ is fed up to the next SGRU. Figure 3.4 illustrates the process of atomic vector construction in SGRU. In our architecture, when the SGRUs are tied, they form the block called kSGRU, where k denotes the number of SGRUs. To obtain the final representation of the molecule from the set of atomic vectors, we apply the sum that runs over all the atoms in the chemical compound as follows:

$$y_a = \sum_{i=1}^{n} aout_i \tag{3.3}$$

3.3.2 Word embedding module and the BiGRU network (block 2)

The word embedding unit

Indeed, learning from SMILES is the second key step in our method. Here, to extract the most significant molecular structural information, we present an approach that includes the word embedding unit and the BiGRU network.

We define a SMILES string as follows: $S = \{e_1, e_2, \ldots, e_M\}$. In other words, at the beginning, a SMILES string is seen as a set of M elements, where e_i refers to the ith element in the set. We assume that the elements can be repeated. The problem of cutting S is solved by a sliding window of size n over S. In each iteration, the window moves to the right by one position. As a result we obtain $S_w = \{slice_1, slice_2, \ldots, slice_L\}$, where $slice_i$ denotes the ith word, $L = M - n + 1$. After processing all SMILES, one obtains all words. Finally, these sequences are used to form molecular word embeddings.

The goal of word embedding is mapping semantic properties of words into a dense vector representation. It enables to capture the latent semantics of the language data. This work employs the Word2Vec [109] vectors pre-trained on Google News to create SMILES substrings embedding. To specify, a feed-forward neural network converts each $slice_i$ of the sequence S_w into a vector v_i. Then, the sequence of embedded vectors $V = v_1, v_2, \ldots, v_L$ is fed up to the next HybNN's component, i.e. the BiGRU network.

The BiGRU network unit

In order to deal with variable-length sequences and provide the integration of contextual information working with SMILES, we adopt the BiGRU network [31]. This component is designed to take the sequence of embedded vectors as input and is expected to return a molecular representation. Suppose, we have $X = \{x_1, x_2, \ldots, x_t\}$, where $x_t \in \mathbb{R}^e$. The functions we implemented can be formulated as below.

$$Z = \sigma(U^z x_t + W^z s_{t-1} + b_z) \tag{3.4}$$

$$r = \sigma(U^r x_t + W^r s_{t-1} + b_r) \tag{3.5}$$

$$h = \psi(U^h x_t + W^h(s_{t-1} \odot r) + b_h) \tag{3.6}$$

$$s_t = (1 - Z) \odot h + Z \odot s_{t-1} \tag{3.7}$$

where $\sigma(\cdot)$ and $\psi(\cdot)$ represent different activation functions, i.e. Sigmoid and Hyperbolic tangent, respectively. U^z, U^r, U^h, W^z, W^r, W^h refer to the corresponding weight coefficients. Z is an update gate to regulate how much the recurrent unit computes its hidden state, while r refers to a set of reset gates. Thus, for r close to 0, the reset gate lets the recurrent unit forget the previous computation state. In addition, $\odot$ specifies an element-wise multiplication. Please also note that s_t of each GRU unit at time t denotes the current hidden state. Similarly, the previous hidden state is represented by s_{t-1}.

The output of BiGRUs at time t is combined with the results of the forward and backward GRUs at the same time. Then, when all the GRU units pick up all information along the forward and backward propagation, the output layer is updated. Therefore, the BiGRU's forward hidden state $\widetilde{h_t^i}$ is computed as follows:

$$\widetilde{h_t^i} = GRU(h_t^{i-1}, \widetilde{h_{t-1}^i}), \tag{3.8}$$

where h_t^{i-1} is the concatenated output obtained from layer $i-1$ in the BiGRU.

This way, the designed approach with learnable ability enables to extract vital features from a SMILES. These features are fed into a dense layer. As a result, we obtain the learned vector y_b that contains relevant structural information connected with the molecule.

3.3.3 Prediction

Finally, for each molecule, a concatenation layer is employed to link the structural feature vector y_b learned based on the SMILES and the physiochemical feature vector y_a learned from the physiochemical properties. More specifically, we obtain a vector $y_{out} = [y_a \cdot y_b]$, where $\cdot$ denotes the concatenation of vectors. Then, the output feature vector is fed into a fully connected layer that returns a vector $out \in \mathbb{R}^2$ as follows:

$$out = \sigma(W_o y_{out} + b_o), \tag{3.9}$$

where W_o refers to the weight matrix, $\sigma(\cdot)$ is Sigmoid activation function, and b_o is the bias vector. The last step is bioactivity prediction. We implemented it using a SoftMax function. Therefore, the function is formulated as follows:

$$p_i = \log\left(\frac{e^{out_i}}{\sum_{k=1}^{2} e^{out_k}}\right), \tag{3.10}$$

where p_i defines the log probability that the molecule with attribute vector out belongs to the ith class (positive or negative), while out_i is the ith element value of the vector out.

3.4 Experiments and Results

3.4.1 Data collection

We use fourteen binary classification datasets in our experiments to test the performance of HybNN. The datasets are derived from the PubChem database [155]. Each dataset includes binary labels on its bioactivity property toward the targets. The datasets are summarized in Table 3.2.

TABLE 3.2: List of assays used for this study.

PubChem AID	Target Name	Actives	Inactives
2358	Protein phosphatase 1, catalytic subunit, alpha isoform 3	1006	934
1915	Group A Streptokinase Expression Inhibition	2219	1017
463213	Identify small molecule inhibitors of tim10-1 yeast	4141	3235
463215	Identify small molecule inhibitors of tim10 yeast	2941	1695
488912	Identify inhibitors of Sentrin-specific protease 8 (SENP8)	2491	3705
488915	Identify inhibitors of Sentrin-specific protease 6 (SENP6)	3568	2628
488917	Identify inhibitors of Sentrin-specific protease 7 (SENP7)	4283	1913
488918	Identify inhibitors of Sentrin-specific proteases (SENPs)	3691	2505
492992	Identity inhibitors of the two-pore domain potassium channel (KCNK9)	2094	2820
504607	Identify inhibitors of Mdm2/MdmX interaction	4830	1412
624504	Inhibitor hits of the mitochondrial permeability transition pore	3944	1090
651739	Inihibition of Trypanosoma cruzi	4051	1324
651744	NIH/3T3 (mouse embryonic fibroblast) toxicity	3102	2306
652065	Identify molecules that bind r(CAG) RNA repeats	2966	1287

3.4.2 Evaluation

We conduct experiments to demonstrate the effectiveness of HybNN from various aspects: 1) predictive performance on validation sets; 2) predictive performance on test sets; 3) analysis how fast a learning machine improves its behaviour; 4) the impact of chemical diversity; 5) the verification of transferability; 6) the influence of varied number of SGRU layers; 7) the impact of the single blocks.

We use mini-batch stochastic gradient descent (mini-batch SGD) with the Adam optimizer [93] to train our HybNN. The batch size is set to 64 and the initial learning rate is $1e^{-5}$. It should also be pointed out that the weights are initialized with the He (Kaiming) initializer [70]. In addition, the hyperparameters that need to be optimized are given Table 3.3. All chemical compound datasets are split into training, validation and testing collections by following the 8/1/1 ratio. Moreover, we use the area under the receiver operating characteristic curve (AUC-ROC) as a performance metric. In addition, the AUC-ROC scores in the tables and figures are obtained by taking the average from the AUC-ROC values of ten independent trials. The significance is evaluated by a one-sided Wilcoxon signed-rank test. The results are statistically significant for a p-value less than 0.05.

The performance of HybNN was compared with four algorithms that have been used in previous studies for molecular properties prediction, including RF [17], Chemception [58], SMILES2vec [59], ChemixNet [118]. The detailed results, including the validation and test sets, are presented in Table 3.4 and Figure 3.5, respectively. In our configuration, we employ RF with 400 trees run on Morgan (ECFP) fingerprints. As shown in Table 3.4, for all datasets except of AID 488912, AID 492992, and AID 652065 HybNN achieves a favorable performance in the validation dataset. For AID 488912, ChemixNet

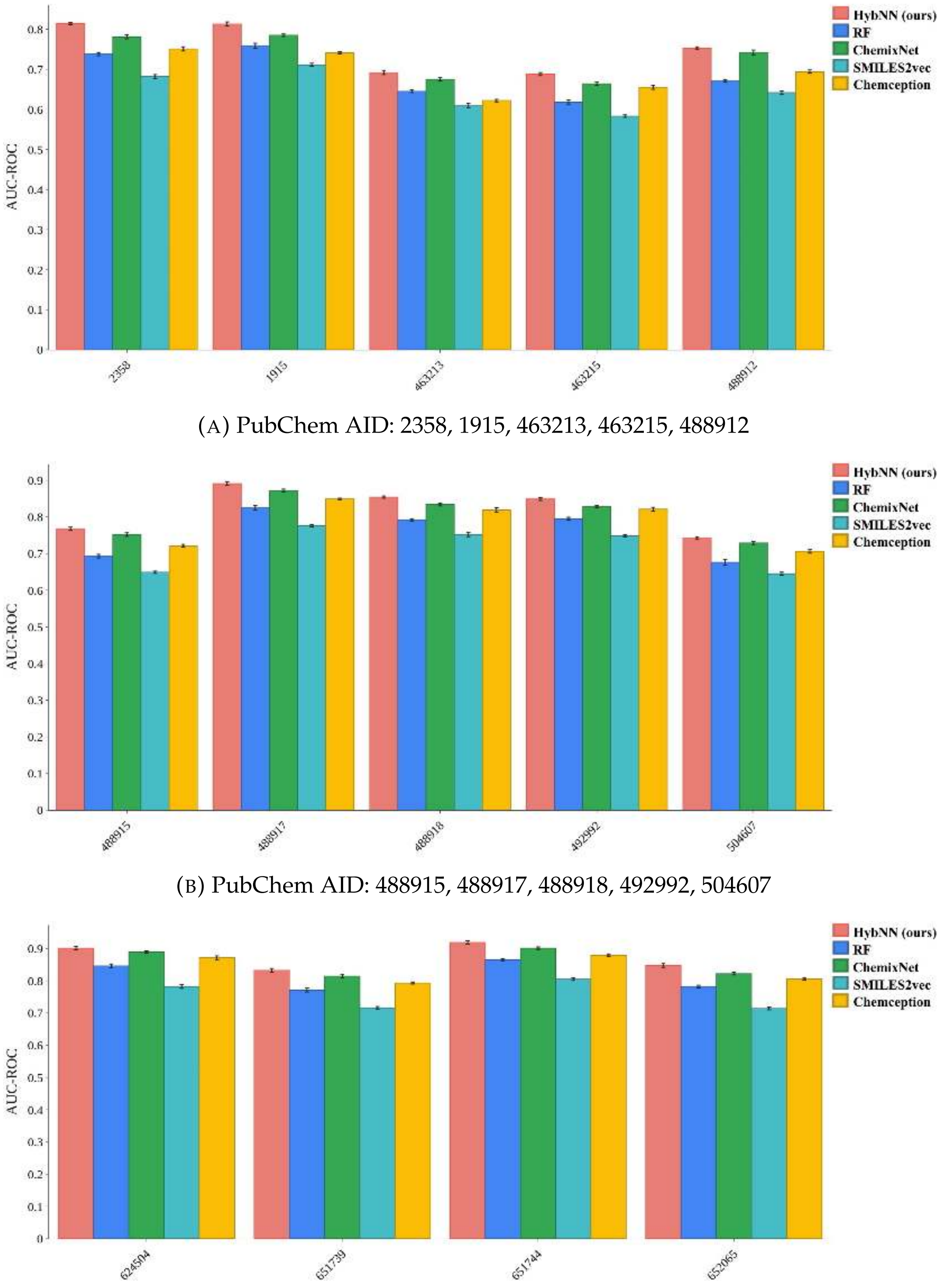
TABLE 3.3: Hyperparameters for HybNN.

Hyperparameters	Values considered
learning rate	$10e^{-3}$, e^{-3}, $10e^{-4}$, e^{-4}, $10e^{-5}$, e^{-5}, $10e^{-6}$, e^{-6},
batch size	8, 16, 32, 64, 128
activation function	ReLU
dropout ratio	0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7, 0.75
number of SGRUs	1-8
weight initialization function	He (Kaiming) [70]
$\|\|y_a\|\|$	150
$\|\|y_b\|\|$	50
number of epochs	37 - 63

beats the other models (0.809) but our approach reveals the second-best performance among the results (0.781). For the AID 492992 and AID 652065 validation datasets, ChemixNet obtains AUC-ROC of 0.851 and 0.889, respectively. In turn, HybNN achieves the second highest score, i.e. 0.847 and 0.873, respectively.

At the same time, the proposed methodology outperforms all benchmark models in all test datasets. Interestingly, our HybNN also obtains the best AUC-ROC scores for AID 488912, AID 492992, and AID 652065 (Figure 3.5). It achieves an AUC-ROC of 0.753 ± 0.003, 0.849 ± 0.004, and 0.847 ± 0.006, respectively. ChemixNet reports the second-best performance with scores 0.742 ± 0.006, 0.828 ± 0.003, and 0.822 ± 0.004. These results fully prove the validity of our HybNN. All comparisons besides AID 2358, AID 463215, AID 504607, AID 652065, AID 488915 and AID 651744 are statistically significant. Therefore, more investigation is needed on this point in the future.

An appropriate parameter setting is a crucial step to successful training deep neural networks. The next experiment investigates the loss function in the training and validation set. The Figure 3.6 indicates that the loss decreases and then stabilizes for both training and validation set. One may observe



(A) PubChem AID: 2358, 1915, 463213, 463215, 488912

(B) PubChem AID: 488915, 488917, 488918, 492992, 504607

(C) PubChem AID: 624504, 651739, 651744, 652065

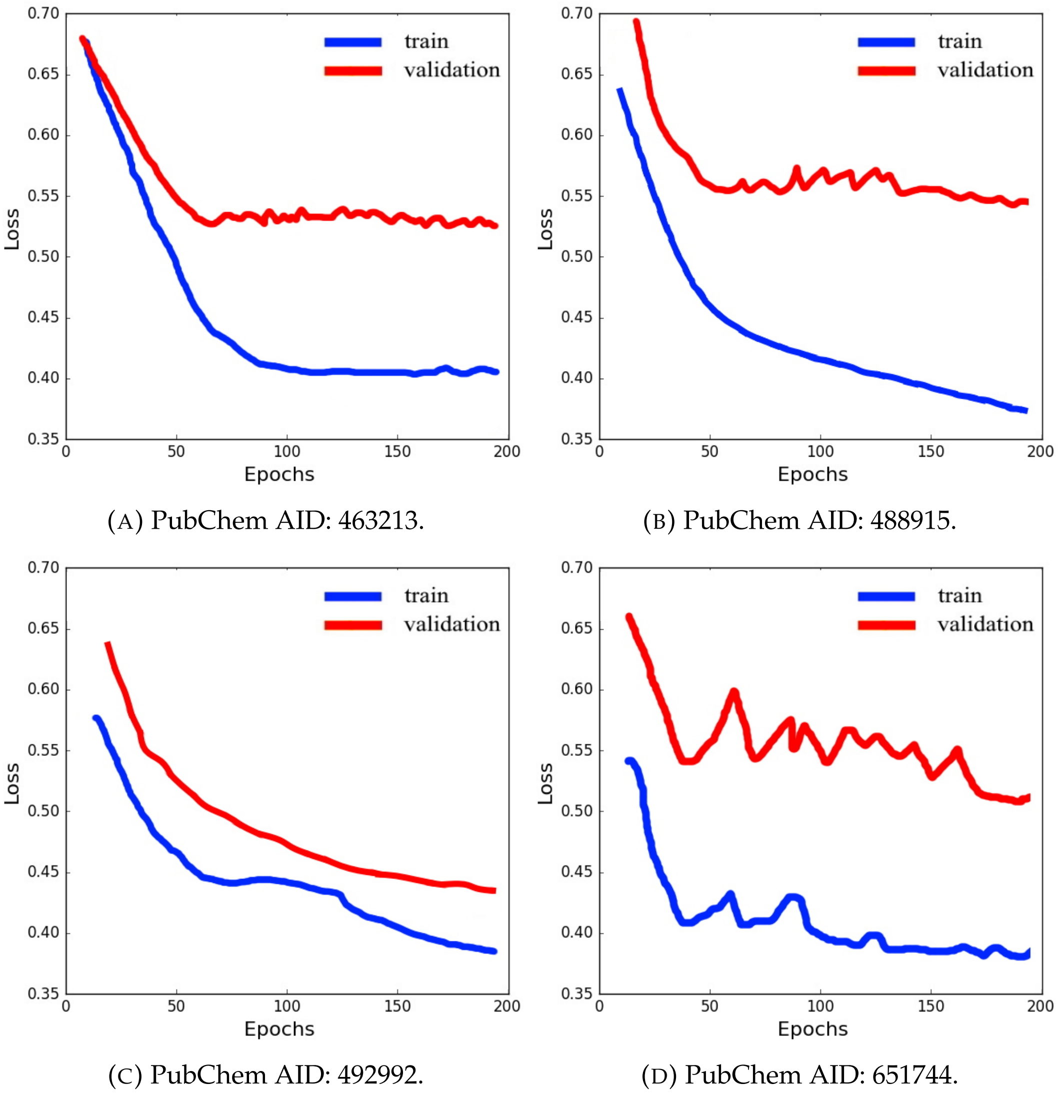
FIGURE 3.5: The performance of HybNN over all datasets. Our model outperforms the other methodologies on all test datasets. Nevertheless, for AID 2358, AID 463215, AID 504607, AID 652065, AID 488915 and AID 651744 the results are not statistically signifcant.

TABLE 3.4: AUC-ROC for validation sets. HybNN achieves a good performance on 11 out of 14 datasets.

PubChem AID	HybNN (ours)	RF	ChemixNet	SMILES2Vec	Chemception
2358	**0.842** ±0.005	0.769 ±0.006	0.809 ±0.005	0.715 ±0.007	0.780 ±0.004
1915	**0.841** ±0.005	0.790 ±0.006	0.813 ±0.008	0.741 ±0.005	0.768 ±0.006
463213	**0.718** ±0.006	0.672 ±0.006	0.703 ±0.005	0.637 ±0.006	0.652 ±0.007
463215	**0.716** ±0.007	0.644 ±0.006	0.692 ±0.006	0.613 ±0.009	0.683 ±0.008
488912	0.781 ±0.004	0.701 ±0.007	**0.809** ±0.008	0.672 ±0.006	0.729 ±0.005
488915	**0.798** ±0.006	0.721 ±0.005	0.780 ±0.007	0.675 ±0.006	0.751 ±0.007
488917	**0.920** ±0.006	0.851 ±0.007	0.899 ±0.006	0.803 ±0.007	0.875 ±0.006
488918	**0.885** ±0.006	0.821 ±0.005	0.855 ±0.007	0.749 ±0.006	0.855 ±0.005
492992	0.847 ±0.004	0.821 ±0.005	**0.851** ±0.006	0.774 ±0.005	0.849 ±0.006
504607	**0.771** ±0.008	0.704 ±0.006	0.754 ±0.006	0.672 ±0.008	0.731 ±0.006
624504	**0.929** ±0.006	0.876 ±0.006	0.915 ±0.007	0.810 ±0.006	0.903 ±0.005
651739	**0.861** ±0.007	0.801 ±0.008	0.842 ±0.006	0.741 ±0.007	0.813 ±0.006
651744	**0.945** ±0.006	0.889 ±0.007	0.925 ±0.007	0.926 ±0.006	0.902 ±0.007
652065	**0.873** ±0.006	0.806 ±0.006	**0.889** ±0.007	0.743 ±0.007	0.831 ±0.007

that the validation loss does not show a descending trend at around 40-60 epochs. Finally, we evaluate the loss on the validation set every epoch and keep the model with the lowest validation loss to avoid overfitting on the training set.

In the next experiments, we verified the effects of different numbers of SGRUs on the performance of HybNN. It may be observed that, depending on the dataset, different numbers of SGRUs lead to favorable performance (see Figure 3.7). We believe, this is related to the complexity of molecular structure



(A) PubChem AID: 463213.

(B) PubChem AID: 488915.

(C) PubChem AID: 492992.

(D) PubChem AID: 651744.

FIGURE 3.6: Loss function.

in datasets and the quality of datasets. For instance, the molecular structure of AID 651744, AID 492992 and AID 488915 datasets is relatively simple, so the model needs more SGRUs and stronger learning ability. The model does not work well on AID 463213 datasets, which may be due to the fact that the data in AID 463213 datasets are not clean enough to cause serious overfitting [130]. Therefore, this may lead to not too many SGRUs in our model.

We also checked if the chemical similarity has a significant connection to the predictive performance of HybNN. Therefore, the prediction error was calculated as a function of the average Tanimoto [126] similarity of the individual molecules to the rest of the training set. In fact, one may observe a correlation between the model error and the average similarity. The prediction accuracy tends to increase as the average similarity increases between the samples for which molecular bioactivity is predicted and the rest of the chemical compounds included in the training dataset.

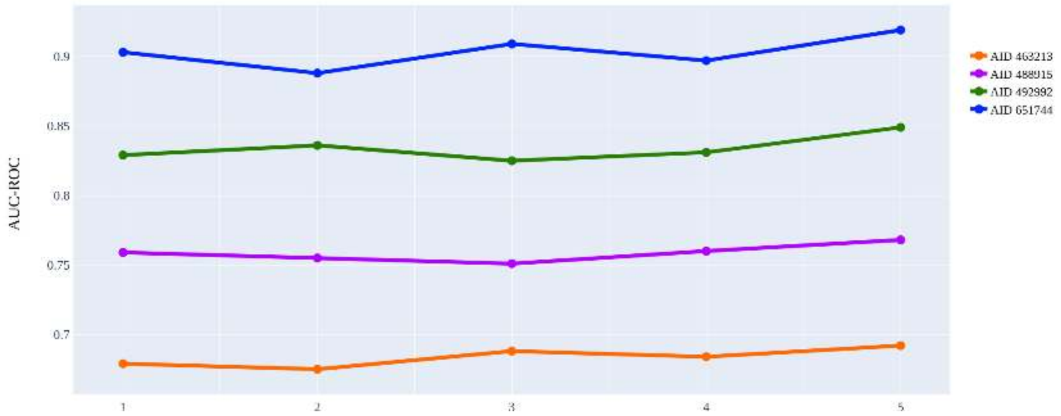


FIGURE 3.7: Performance vs. different number of SGRUs. It may be observed that for different datasets, different numbers of SGRUs lead to varying performance.

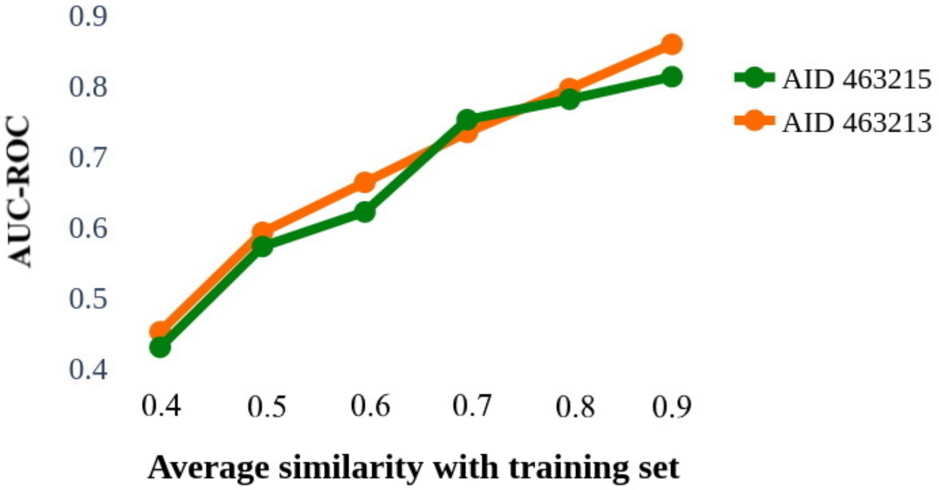


FIGURE 3.8: Prediction accuracy (measured as AUC-ROC) versus the average similarity of the molecule to the rest of the training set. The plot indicates a correlation between model error and average similarity.

Furthermore, we conduct ablation studies on the sampler architecture.

- The variant which only learns the representation using block I.
- The variant which only learns the representation using block II.

The ablation experiments results on both tasks are shown in Figure 3.9. There are several findings from this figure. First, the outcomes indicate that both block I and block II are all valuable for the prediction task. Second, among two variants of HybNN, the version where block I is removed has the worst performance. In case of AID 463213 dataset, the variant that has no block I reveals AUC-ROC of 0.641, whereas the AUC-ROC of 0.660 is obtained by the version that has no block II. Similarly, for AID 504607, AID 463215, AID 492992, and AID 651744, we report the scores 0.702, 0.621, 0.784, 0.843 when block I is removed, while the AUC-ROC of 0.731, 0.663, 0.827, 0.901 when block II is omitted. It may suggest that employing regular one-dimensional convolution to process the spatial graph matrix of the molecule is a key ingredient in HybNN architecture.

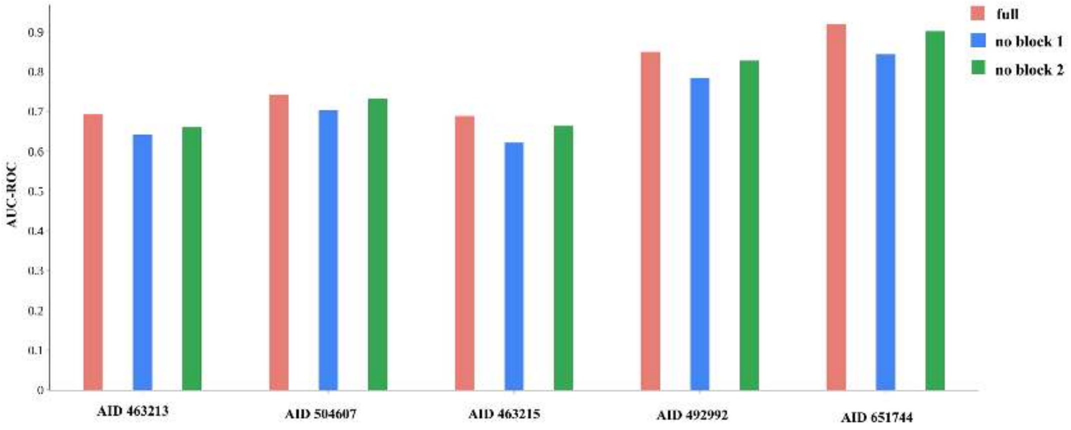


FIGURE 3.9: Performances of different model variants for ablation study. Firstly, both blocks are valuable for the predictive performance. Secondly, the variant that has no block I has the worst performance. It indicates that regular one-dimensional convolution is beneficial here.

3.5 Conclusion

The goal which motivates this chapter, will be to leverage both graphs and sequences to learn effective representations of molecules for the drug bioactivity prediction task. Therefore, we introduce a deep learning-based architecture, called HybNN, for learning representations that integrates specifically designed concepts, i.e. a stack of convolutional layers, and an RNN based on bidirectional gated recurrent unit (BiGRU). Therefore, the goal is to obtain an end-to-end molecular representation and improve chemical bioactivity prediction results. Our method automatically learns a mixed molecular representation from both physiochemical properties and SMILES contextual information that describes the structure of the chemical compound. The performance of this methodology was compared with four state-of-the-art models including RF, ChemixNet, SMILES2vec and Chemception. The superiorities and competitiveness of HybNN are demonstrated by extensive experimental outcomes.

Chapter 4

Learning A Fragment-Oriented Representation for Supervised Learning Problems

In this study, we present Subgraph Encoded Neural Network (SENN), the structure-based deep neural network, designed to predict the toxicity of small molecules for drug discovery applications. The locally exploited graph structure allows the algorithm to model the complex phenomenon of molecular properties and affects the interpretability of our system. Furthermore, by incorporating global molecular features, such as a molecular weight, SENN is able to predict new toxic molecules. SENN shows satisfactory results on a widely used benchmark achieving an AUC-ROC greater than the previous methodologies (RF [17], SVM [153], GIN [169], GCNN [4], Weave [88]) for four datasets and also surpassing the other approaches (GCNN [4], TopTox [166], Weave [88], feed-forward neural network (FFN)) for two datasets in case of regression task.

4.1 Introduction

In the previous Chapter we addressed the problem of classification and applied our solution to a molecular bioactivity prediction task. We have discussed how information about the physiochemical properties and structure of the chemical compound can complement and enrich the molecular representation and affect the model performance. However, another challenge in machine learning and deep learning is the sample size. Overall, the problem is as follows: training a learning model needs enough data to prevent overfitting and extract valuable insight to learn representations. As a consequence, when one has not sufficient data for a learning task, it is hard to use the model and make the model interpretable. Therefore, now it is time to tackle the challenge of representation learning when the dataset is quite small and biased. In the second contribution, we cast this problem into a problem involving molecular data where the task is to predict molecular properties. To specify, our next goal, which motivates this chapter, will allow to exploit the molecular structure more carefully than the previous approaches since we consider the fragments of the chemical compound. Here, we focus on the toxicity prediction, both as a classification and as a regression task.

Indeed, drug development is considered as a fine balance of optimizing drug-like properties that aim to maximize safety, efficacy, and pharmacokinetics. Moreover, many studies indicate that poor toxicity remains the major limiting aspect of drug discovery [69]. One strategy that has been widely employed is *in vivo* methodology. However, time-consuming wet-lab experiments or simulations result in a limited number of chemical compounds with validated properties [67]. In addition, it happens that they do not necessarily scale between animal models and humans. To address these issues, there has recently been a shift towards *in vitro* and to machine learning based *in silico* techniques. Thus, numerous *in silico* approaches for predicting toxicity properties of molecules have been developed [101]. They range from data-based methods such as quantitative structure-activity relationship (QSAR), similarity searches [26], structure-based approaches [152, 2], to using algorithms including Random Forests (RF) [120], and Support Vector Machines (SVM) [23].

To foster further development and to better understand the underlying mechanisms of action of various toxic chemicals, we propose **S**ubgraph **E**ncoded **N**eural **N**etwork (SENN) that investigates the role of atoms connections in the molecular graph and global molecular features. Summarizing, our contributions are the following.

- We primary focus on the challenge of exploiting a graph structure. To this aim, we propose *k-paths subgraphs*, where a parameter k is used to extract subgraphs from a single input graph.
- A set of global molecular features such as a molecular weight or a number of rotatable bond is selected to increase the model's discriminative capability.
- We report objective results against state-of-the-art methodologies. For instance, on the AR toxicity dataset, our SENN obtains significantly better AUC-ROC score (0.802 ± 0.006) than that by the well-known prediction approaches including GCNN, SVM, and GIN.

4.2 Related Work

A promising way of extracting relevant patterns from the data to detect toxic chemical compounds is using the concept of deep learning [85, 145]. Since a main target of toxicological research are the set of steric and electronic fragments that together produce a certain toxicological effect [91], deep learning architectures seem to be well suited for this task. For this reason, a wide range of papers have explored methods such as Weave [88], TopTox [166], GIN [169], GCNN [4] working directly on the molecular graph structure, leading to state-of-the-art performance. In Weave, the Kearnes et al. propose a novel featurization approach that encodes both the local chemical environment and the connectivity of atoms in a chemical compound [88]. The authors pay attention to connectivity that is represented by more detailed pair features instead of just a simple neighbor listing. The dataset collection used in the work contains over 38 M data points and includes targets from many different biological classes. In fact, the experiments with graph convolution models do not beat all fingerprint-based methods. However, they demonstrate that graph convolution models with optimizations may exceed the performance of the best available fingerprint-based approaches. In turn, Wu and Wei introduced an element specific topological descriptor together with a set of extra descriptors based on established physical models and integrated them with a variety of advanced machine learning methodologies [166]. The method leads to an improvement over current state-of-the-art outcomes on a dataset related to the prediction of small molecular quantitative toxicity. Next, Xu et al. proposed GIN, a GNN architecture under the neighborhood aggregation framework [169]. The method was evaluated on bioinformatics datasets and social network datasets. The results reveal that GIN beats or achieves comparable performance as the other state-of-the-art variants of GNNs. Another recent method is proposed by Altae-Tran et al. The authors developed an iterative refinement long short-term memory (LSTM) that is seen as a modification of the matching-networks architecture and the residual convolutional network for predicting chemical property [4]. It is worthy to note that their algorithm brings significant improvement on a wide range of datasets meaningful for drug discovery, such as toxicity prediction or adverse reactions detection. However, one of the key limiting factors in the successful deployment of these approaches for toxicity prediction is the requirement for large datasets from which to train such complex models.

Research groups have been also approaching toxicity challenges through a variety of machine learning techniques such as RF, SVM, and more. Polishchuk et al. employes RF to QSAR analysis of aquatic toxicity of chemical compounds [120]. Their model's quality is assessed on two external test datasets of 110 and 339 molecules. The results obtained indicate that RF can be regarded as a very promising approach since it has comparable or better statistical characteristics than the corresponding methodologies. On the other hand, the group of Cao et al. constructed a SVM-based toxicity detection system for classifying five toxicity datasets [23]. The method measures the similarities of chemical compounds by the substructure or fragment information hidden in the SMILES format. The results reveal that most toxic behaviour is usually associated with structural attributes. Nevertheless, although the available models are very helpful for drug design, more accurate approaches can be developed for toxicity prediction.

4.3 Methodology

4.3.1 Problem formulation

The core input data structure considered throughout this methodology is the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a way to encode molecular data. As alluded to earlier, $\mathcal{V}$ denotes a set of atoms with $|\mathcal{V}| = n$. Here, the graph is seen as a complete undirected graph. Thus, it is assumed that all atoms have interactions with others. This implies that the set of edges can be expressed as $|\mathcal{E}| = \frac{n(n-1)}{2}$. Furthermore, in this setting, each edge $\mathcal{E}$ is associated with two types of attributes: edge type and spatial information. However, in molecules, there are very few types of chemical bonds. It is therefore reasonable for simplicity to assume that each of these bonds $e_{ij} \in \mathcal{E}$ is associated with a parameter

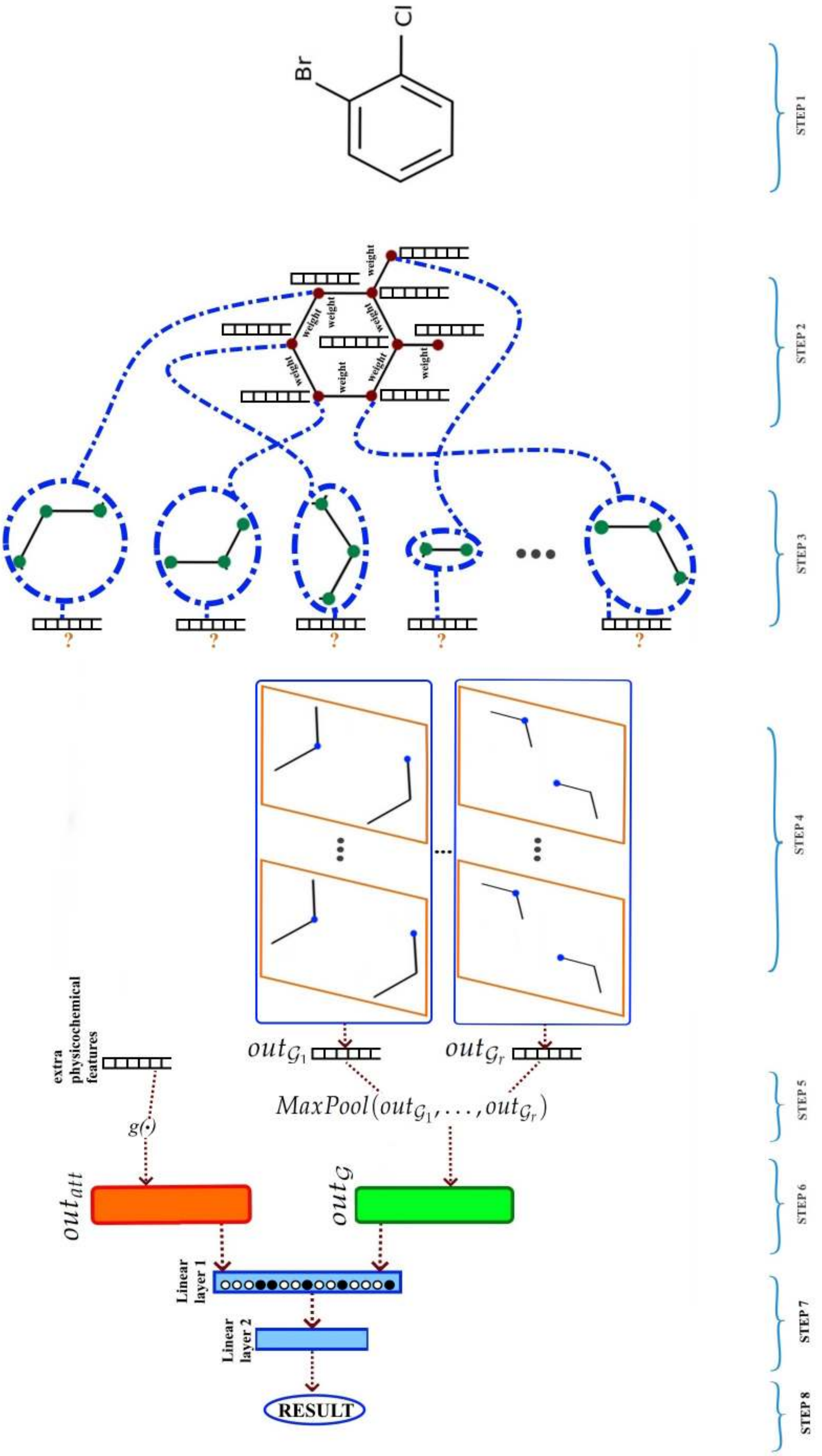


FIGURE 4.1: Overview of SENN. Our approach can be split into seven steps. (1) Input. (2) Molecular attributes assignment. (3) Subgraphs construction. (4) Graph convolution. (5, 6) Embedding. (7) Features concatenation. (8) Prediction.

$weight_{ij} \in (0, 1 > \in \mathbb{R}$. Throughout this chapter, our algorithm also operates on extra physicochemical $features$ that describe the general properties of the molecule. Depending on the task, our target is to construct a regressor or a classifier to predict the toxicity value of molecules or to determine whether the chemical compound is toxic or not, respectively. Formally, the problem is defined as per Equation 4.1:

$$\psi(\phi(\mathcal{G}), g(features)) = y, \tag{4.1}$$

where y is the target property to predict, i.e. toxicity. If $y \in \mathbb{R}$ then we focus on a regression task. A binary $y \in \{0, 1\}$ indicates a binary classification task. Moreover, the middle function $\phi : \mathcal{G} \rightarrow \mathbb{R}^d$ is used to learn a molecular graph vector representation and function $g : features \rightarrow \mathbb{R}^f$ aims at learning a dense representation of features. Then, ψ is used to convert the obtained features to the final result.

4.3.2 Subgraphs encoding

Here, we introduce a subgraph encoding that is one of the key contributing factors in our SENN. This idea is motivated by our previous work [158], where we uncovered the importance of random walks on graphs. Generally, for a graph $\mathcal{G}$, our goal is to associate the atoms and bonds of a molecule with a d-dimensional real-valued vector space. In order to meet the challenge of exploring a more efficient way to represent small chemical compounds, an approach that we call *k-paths subgraphs* is employed. To specify, a value of parameter k indicates how many neighboring vertices and edges should be considered, looking from the current vertex, to form a graph. Intuitively, k aims to affect the sensitivity of the final embeddings. In other words, the lower k, the more meaningful the representation should be.

In our approach, we denote a set of all neighboring vertex indices within no more than k edges from the jth vertex by $\widetilde{N}(j,k)$. Obviously, $\widetilde{N}(j,0) = j$. Formally, given the vertex v_j, the k-path subgraph is defined as follows:

$$\mathcal{G}_{j,k}(\mathcal{V}_{j,k}, \mathcal{E}_{j,k}), \tag{4.2}$$

where

$$\mathcal{V}_{j,k} = \{v_i | i \in \widetilde{N}(j,k)\} \tag{4.3}$$

and

$$\mathcal{E}_{j,k} = \{e_{xy} \in \mathcal{E} | (x,y) \in \widetilde{N}(j,k) \times \widetilde{N}(j,k-1)\}. \tag{4.4}$$

Thus, $v_{j,k}$ is called the k-path source vertex in which outgoing paths are given. Consequently, the k-path subgraph for the edge e_{ij} is formulated as:

$$\mathcal{E}_{(ij),k} = (\mathcal{V}_{i,k-1} \cup \mathcal{V}_{j,k-1}, \mathcal{E}_{i,k-1} \cap \mathcal{E}_{j,k-1}). \tag{4.5}$$

4.3.3 Architecture of SENN

Model design

The entire architecture of SENN could be split into eight parts in a high-level discussion (see Figure 4.1). The initial input to the SENN is a graph $\mathcal{G}$ that represents a molecule (step 1). Thus, at the beginning, we assign an atom attributes to each vertex and each edge is also associated with its weight (step 2). For example, the weight may equal the normalized multiplicity of the bond it refers to. Then, the graph $\mathcal{G}$ is preprocessed to obtain *k-paths subgraphs*, $S = \{\mathcal{G}_{1,k}, \mathcal{G}_{2,k}, \ldots, \mathcal{G}_{r,k}\}$, where r denotes the total number of subgraphs (step 3). Each subgraph $\mathcal{G}_{i,k}$ is fed into a graph convolutional neural network (step 4) and the embedding $out_{\mathcal{G}_i} \in \mathbb{R}^t$ is returned (step 5 in Figure 4.1 and Equation 4.8). To obtain the output $out_{\mathcal{G}} \in \mathbb{R}^d$ from the set of vectors associated with subgraphs embedding, we use the max pooling operation and combine the obtained vectors by concatenation along the feature dimension.

At the same time, extra physicochemical features are propagated through a multilayer perceptron (MLP) block. In consequence, the chemical features are embedded into a continuous vector space, $out_{att} \in \mathbb{R}^f$.

In the next step, the vector $out_{\mathcal{G}}$ is concatenated with the vector $out_{att} \in \mathbb{R}^f$. The combined representations form a feature vector, $out_{con} \in \mathbb{R}^{d+f}$, defined as

$$out_{con} = out_{\mathcal{G}} || out_{att} \tag{4.6}$$

that is the input to a few linear layers with a dropout (step 7), and a final task layer (step 8). In this setting, the final hidden layer is seen as the learned representation $out_{\mathcal{M}}$.

Transition strategy

However, as it was mentioned, before the final task is performed, we deal with the graph embedding operation and GCN. Therefore, firstly, for a given graph $\mathcal{G}$, all distinct k-path subgraphs are extracted (step 3). Then, a random unit-norm vector is associated with each subgraph. And from that time on, the embeddings assigned to vertices are updated by GCN's layers. Specifically, each vector is replaced with the average over all vectors in its neighbourhood. Next, a linear transformation is applied. As a result, the computed vertex embeddings are averaged, and we obtain a t-dimensional graph representation.

More formally, let us assume that a subgraph $\mathcal{G}_{i,k}$ has an adjacency matrix A with m vertices. Additionally, suppose that $Z^* \in \mathbb{R}^{m \times t}$ denote the embedding of the vertices. Hence, we can define the transition function as per Equation 4.7. More precisely, given an input subgraph $\mathcal{G}_{i,k}$ with adjacency matrix A consisting of m nodes (atoms), and the quantity $X^{(0)} \in \mathbb{R}^{m \times t}$ representing the t-dimensional embedding of the nodes, an l-layer GCN updates node embeddings using the following transition function:

$$Z^{(j+1)} = ReLU(\tilde{A}W^{(j)}Z^{(j)} | \forall j \in \{0, 1, \ldots, l-1\}), \tag{4.7}$$

where l refers to the GCN layer, $\tilde{A} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$ denotes the normalized adjacency matrix. Moreover, $\hat{A} = A + I$ and $\hat{D}$ refers to the degree matrix of $\hat{A}$. $W^{(j)} \in \mathbb{R}^{t \times t}$ is the weight-matrix of the jth-layer of the GCN.

Furthermore, as we mentioned earlier, the calculated embeddings from the last layer of GCN are averaged, and the subgraph representation can be defined as follows:

$$out_{\mathcal{G}_i} = \frac{1}{m}(\sum_{i=1}^{m} Z^{(l)}[i,:])^T. \tag{4.8}$$

SENN's architecture has several compelling advantages. However, the significant advantage includes interpretability, since it benefits both from bottom-up and top-down approaches. Here, the bottom-up methodology is associated with k-paths subgraphs, where one can focus on fragments of the graph. In turn, selection of global molecular features may be seen as the top-down approach.

4.4 Experiments and Results

All models were trained using the stochastic gradient descent (SGD) algorithm with the ADAM optimizer [93]. The initial learning rate was randomly chosen from $5e^{-5}$ to $5e^{-4}$. Different seeds were selected for all models to verify the robustness of the models, and grid search was utilized for hyper-parameter screening. In addition, the extra physicochemical features employed in step 5 are connected with the selected general attributes of chemical compounds. They include the features extracted by ChemoPy [24] such as a molecular weight or a number of rotatable bonds. In the experiments, we compared our model with five state-of-the-art models, including both traditional machine learning approach and deep learning methodology, i.e. RF [17], SVM [33], GIN [169], Weave [88], and GCNN [4]. We evaluate statistical significance using a one-sided Wilcoxon signed-rank test. In addition, we claim that the results are statistically significant for a p-value less than 0.05. Here, in Subsection 4.4.1 we provide more insight into the datasets. Then the results of the comparison of our algorithm and other models are given. The performance of the classifiers is evaluated by calculating the AUC-ROC. In turn, the performance of created regression models was evaluated by the root mean square error (RMSE) defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where n refers to the number of samples, y_i is the observed value for the ith observation, and $\hat{y}_i$ is the predicted value.

Furthermore, the means and standard deviations of the AUC-ROC and RMSE scores are measured by five independent trials.

TABLE 4.1: List of assays used for this study.

Dataset Name	**Description**	**Total (unique molecules)**
AR	androgen receptor	9946 (8052)
AhR	aryl hydrocarbon receptor	8713 (7260)
AR-LBD	androgen receptor, luciferase	9105 (7433)
Aromatase	cytochrome P450 enzymes	7654 (6494)
ATAD5	genotoxicity indicated by ATAD5	9635 (7800)
ARE	nuclear factor (erythroid-derived 2)-like 2 antioxidant responsive element	7635 (6427)
ER	estrogen receptor alpha	8227 (6864)
ER-LBD	estrogen receptor alpha, luciferase	9327 (7712)
HSE	heat shock factor response element	8684 (7151)
MMP	mitochondrial membrane potential	7796 (6417)
p53	DNA damage p53 pathway	9172 (7469)
PPAR-gamma	peroxisome proliferator-activated receptor gamma	8718 (7141)

4.4.1 Data collection

In order to objectively demonstrate the advantages of our SENN, multiple toxicity-related datasets are adopted for regression and classification tasks.

Classification task

The dataset was taken from the Tox21 Data Challenge [3] in both SDF and SMILES formats. The data consists of approximately 12 000 compounds and includes twelve different sub-challenges/tasks. Each sub-challenge relates to the prediction of a different type of toxicity. They are grouped into two panels: the “Nuclear Receptor Signaling Panel” (seven assays) and the “Stress

TABLE 4.2: List of assays used for this study.

Dataset Name	Total
LC50	823
LD50	7413
IGC50	1792
LC50-DM	353

Response Panel" (five assays). All assay endpoints are reported in Table 4.1. Please note that the Tox21 set contains duplicated records, i.e. the same SMILES representation in spite of the different chemical compound name. To identify these molecules, the Online CHEmical database and the Modeling environment platform [140] were used. The tool allows us to calculate the INCHI [82] key structure hash to compare structures.

Regression task

We performed extra experiments using four regression-based toxicity data sets [166]. These datasets, namely 96 h fathead minnow LC50 dataset (LC50 set), 48 h Daphnia magna LC50 dataset (LC50-DM set), 40 h T. *pyriformis* IGC50 dataset (IGC50 set) and oral rat LD50 dataset (LD50 set), are presented in Table 4.2.

4.4.2 Performance on the classification task

The detailed results, including the validation sets, are shown in Table 4.3.

Regarding the classification task, we compare the prediction performance of our model (SENN) with the five benchmark models (RF [17], SVM [153] with an RBF kernel, and Gradient Boosting (GB) [51] as implemented in Scikit-learn [119], GIN [169], GCNN [4], Weave [88]). RF is run with 600 trees using Morgan (ECFP) fingerprints. We measured the AUC-ROC scores of the test sets to evaluate the prediction accuracy. It can be observed that the AUC-ROC scores of the validation sets on all datasets are higher than those of the

TABLE 4.3: AUC-ROC for validation sets. Our methodology beats the other approaches in four cases.

Dataset	SENN (ours)	RF	SVM	GIN	GCNN	Weave
AR	**0.816** ±0.007	0.797 ±0.006	0.772 ±0.005	0.773 ±0.006	0.803 ±0.007	0.795 ±0.006
Aromatase	0.837 ±0.007	0.835 ±0.006	**0.865** ±0.005	0.829 ±0.007	0.851 ±0.006	0.832 ±0.007
ER-LBD	**0.829** ±0.005	0.820 ±0.005	0.806 ±0.006	0.792 ±0.007	0.819 ±0.006	0.823 ±0.005
PPAR-gamma	0.830 ±0.008	**0.869** ±0.007	0.834 ±0.007	0.818 ±0.006	0.824 ±0.006	0.806 ±0.005
AhR	**0.935** ±0.007	0.903 ±0.006	0.888 ±0.007	0.901 ±0.006	0.913 ±0.007	0.895 ±0.006
AR-LBD	0.856 ±0.007	**0.891** ±0.008	0.860 ±0.009	0.844 ±0.007	0.871 ±0.006	0.861 ±0.007
ER	0.735 ±0.006	**0.758** ±0.007	0.732 ±0.006	0.702 ±0.007	0.754 ±0.006	0.731 ±0.005
ARE	0.705 ±0.008	**0.832** ±0.006	0.821 ±0.008	0.803 ±0.006	0.817 ±0.006	0.793 ±0.007
p53	0.834 ±0.006	**0.883** ±0.007	0.861 ±0.006	0.820 ±0.008	0.859 ±0.006	0.816 ±0.006
MMP	**0.939** ±0.008	0.902 ±0.009	0.906 ±0.008	0.891 ±0.006	0.911 ±0.009	0.902 ±0.006
HSE	0.789 ±0.007	0.805 ±0.007	**0.827** ±0.009	0.751 ±0.005	0.794 ±0.006	0.773 ±0.007
ATAD5	0.855 ± 0.006	**0.886** ±0.007	0.841 ±0.007	0.828 ±0.007	0.842 ±0.008	0.798 ±0.006

test sets. As Table 4.3 reveals, SENN beats the competitive method in four cases. Our model achieves an average AUC-ROC of 0.816 for AR (the second highest score of 0.803 by GCNN), 0.829 for ER-LBD (the second highest score of 0.823 by Weave), 0.935 for AhR (the second highest score of 0.913 by GCNN), and 0.939 for MMP (the second highest score of 0.911 by GCNN).

Figure 4.2 shows the mean AUC-ROC scores and standard deviations on the test sets for all datasets. Here, SENN achieves the highest score in four cases. More precisely, for AR, ER-LBD, AhR and MMP our approach gets the AUC-ROC of 0.802 ± 0.006, 0.813 ± 0.004, 0.902 ±, and 0.906 ± 0.004, respectively. In turn, the second-best model achieves a score of 0.793 ± 0.004 (GCNN),

TABLE 4.4: RMSE for validation sets (lower is better). Our method achieves slightly better average RMSE scores on both the IGC50 and LD50 datasets.

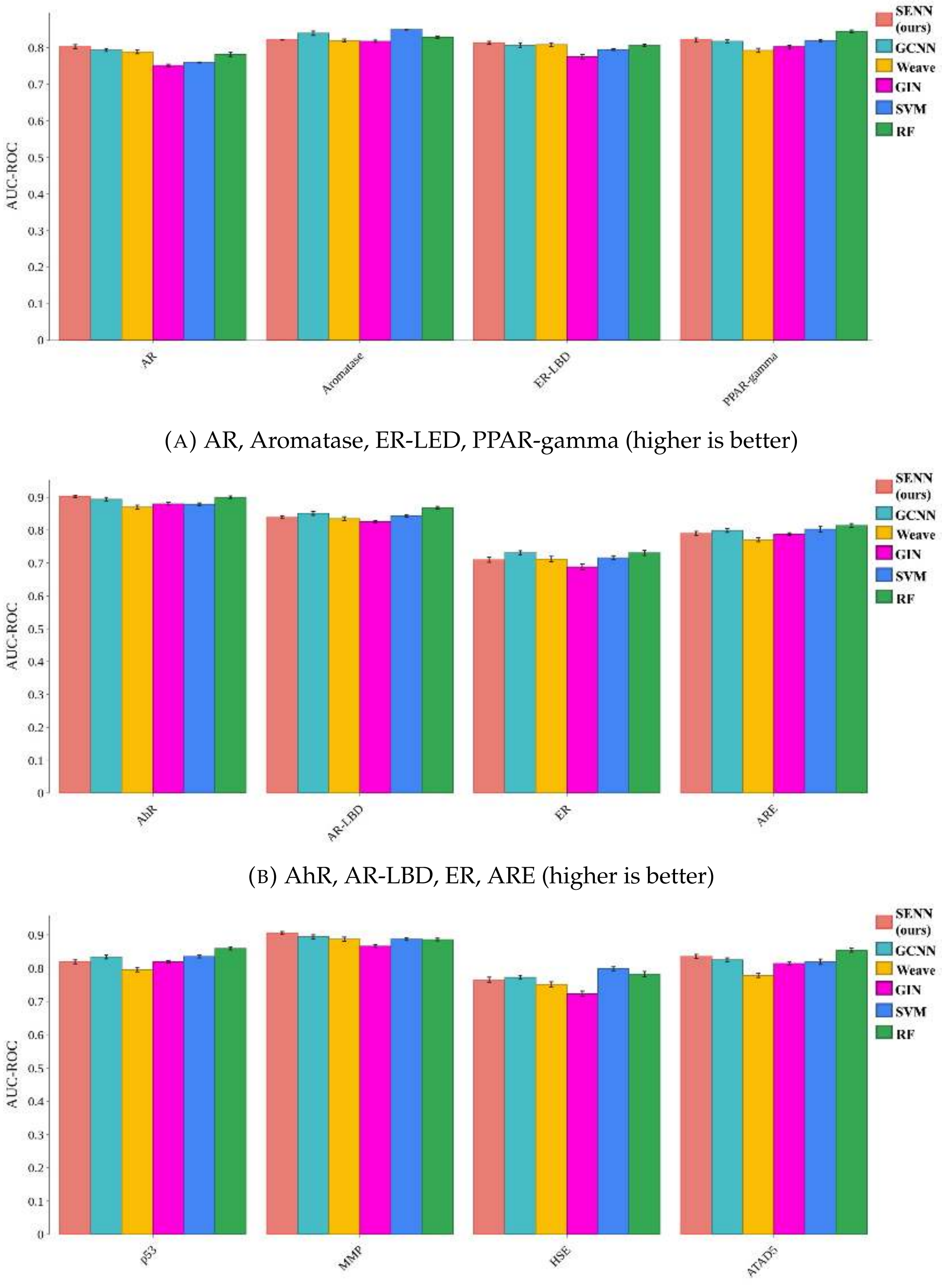
Dataset	SENN (ours)	FFN	TopTox	GCNN	Weave
IGC50	**0.452** ±0.007	0.498 ±0.007	0.461 ±0.008	0.518 ±0.006	0.504 ±0.006
LC50	0.953 ±0.006	0.890 ±0.006	**0.694** ±0.009	0.993 ±0.005	0.871 ±0.005
LC50DM	0.882 ±0.005	0.915 ±0.008	**0.839** ±0.006	0.990 ±0.007	0.848 ±0.008
LD50	**0.589** ±0.008	0.647 ±0.006	0.592 ±0.006	0.641 ±0.007	0.653 ±0.006

0.806 ± 0.007 (GCNN), 0.900 ± 0.004 (RF), and 0.894 ± 0.006 (GCNN), respectively. Please note that only the results on p53, AhR, HSE and ATAD5 are not statistically significant.

4.4.3 Performance on the regression task

For the regression tasks, the prediction performance of SENN is compared with those of four other neural network-based methods: GCNN [4], TopTox [166], Weave [88], and the feed-forward neural network (FFN). The detailed results, including the validation sets, are described in Table 4.4. Undoubtedly, SENN achieves the best RMSE scores on both the IGC50 and LD50 datasets, i.e., 0.452 and 0.589, respectively. Interestingly, TopTox beats all the models for LC50DM (RMSE of 0.839) but our approach outperforms GCNN (RMSE of 0.882 vs. 0.990).

Figure 4.3 shows the RMSE and MAE scores on the test sets against the IGC50, LC50, LC50DM and LD50 datasets. In case of IGC50 and LD50, our approach reveals the best performance among the tested models, i.e. RMSE of 0.415 ± 0.001, MAE of 0.301 ± 0.002, and RMSE of 0.557 ± 0.002, MAE of 0.431 ± 0.003, respectively. The results of the second-best approach are as follows: RMSE of 0.436 ± 0.001, MAE of 0.305 ± 0.001 (TopTox), RMSE of 0.568



(A) AR, Aromatase, ER-LED, PPAR-gamma (higher is better)

(B) AhR, AR-LBD, ER, ARE (higher is better)

(C) p53, MMP, HSE, ATAD5 (higher is better)

FIGURE 4.2: The mean AUC-ROC scores and standard deviations on test sets for all datasets. SENN achieves the highest score for AR (0.802 ± 0.006), ER-LBD (0.813 ± 0.004), AhR (0.902 ± 0.005), and MMP (0.906 ± 0.004). Only the results on p53, AhR, HSE and ATAD5 are not statistically significant. To sum up, the high quality of our model makes it suitable for deployment in leading edge toxicological research.

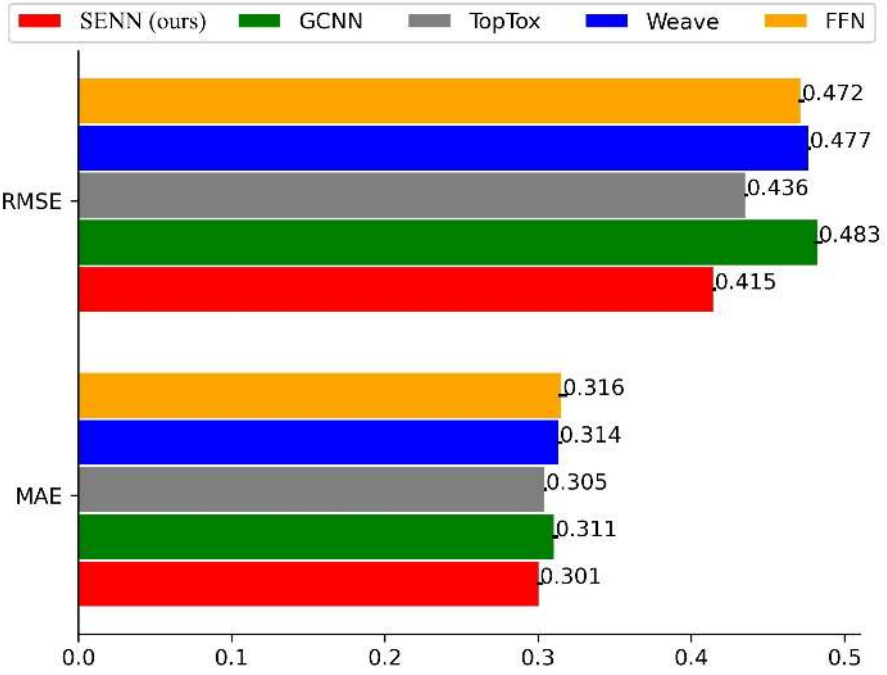
± 0.001, MAE of 0.421 ± 0.002 (TopTox). Moreover, although for LC50DM SENN performs worse than TopTox (RMSE of 0.807 ± 0.002, MSE of 0.593 ± 0.003) and Weave (RMSE of 0.822 ± 0.001, MSE of 0.675 ± 0.003), it is still better than GCNN (RMSE of 0.973 ± 0.003, MSE of 0.942 ± 0.002). It shows that all comparisons besides LC50 and LC50DM are statistically significant.

4.4.4 $\mathcal{L}$ loss

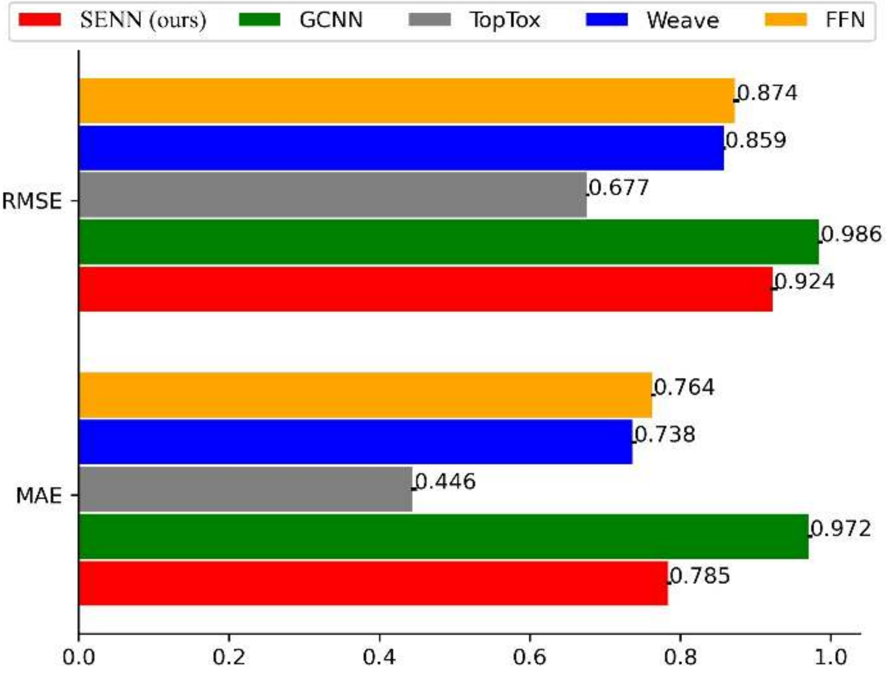
We run our training for a total of 100 epochs. Figure 4.4 shows loss curves during training with the maximum training epoch set to 100 for AhR, Aromatase, ATAD5 and p53. Depending on the dataset, gaps between training and the test may be observed which suggest the presence of overfitting. The largest gap occurred for Aromatase (after 22 epochs) and ATAD5 (after 18 epochs). As illustrated, the overfitting issue observed above was eliminated within 100 epochs.

4.4.5 Similarity validation

In order to systematically investigate the dependence of the prediction error on the chemical compounds similarity, we plotted the prediction error as a function of the average Tanimoto similarity of the individual molecules to the rest of the training set (shown in Figure 4.5 and Figure 4.6). A correlation between model error and average molecules similarity is observed. The prediction error increases when the average similarity decreases between the molecules, for which the compound score is predicted and the rest of the training set molecules. This observation is related to the *Similar Property Principle* which states that similar compounds are likely to have similar properties [163].

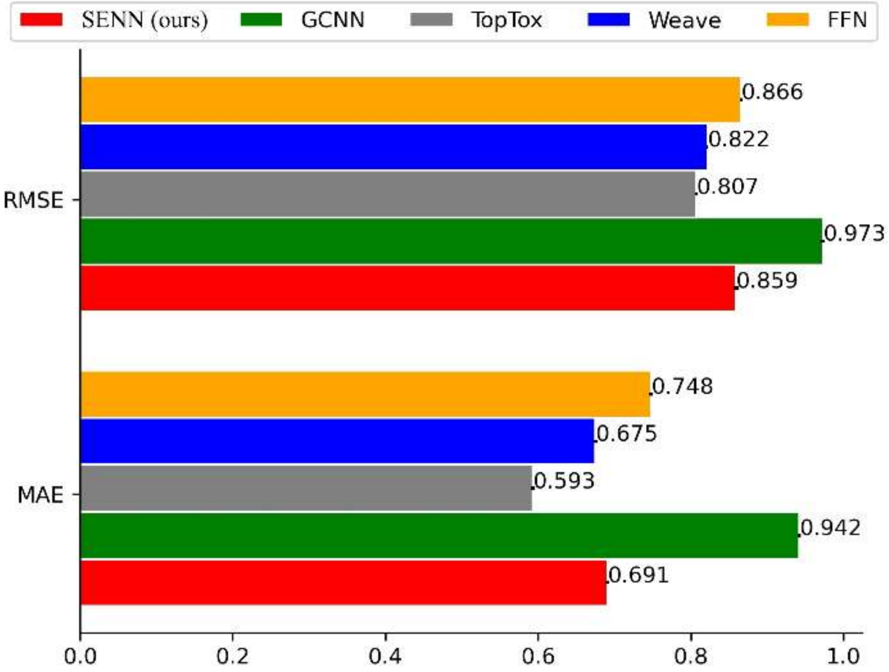


(A) IGC50

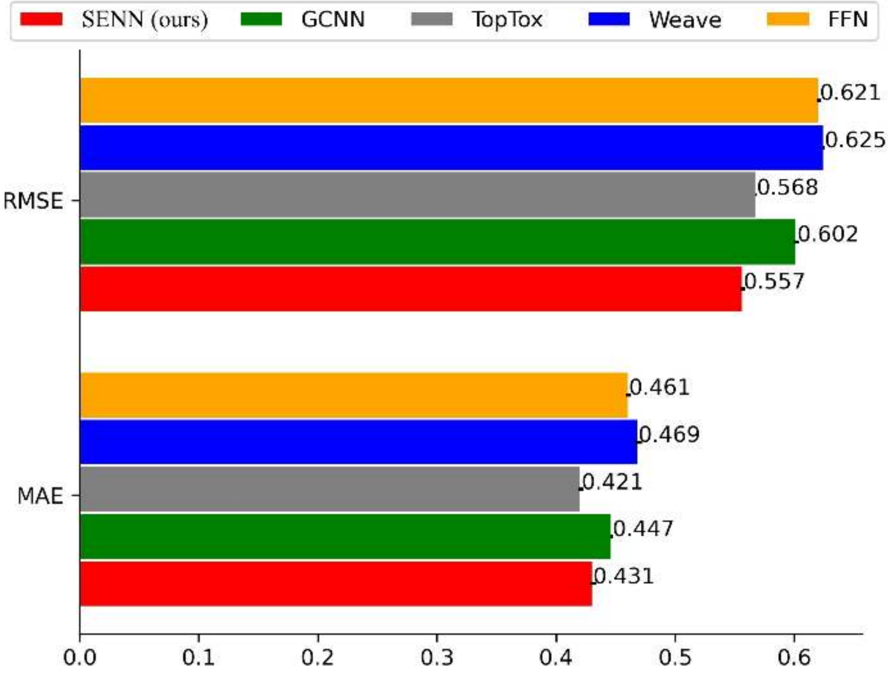


(B) LC50

FIGURE 4.3: The RMSE and MAE scores on the test sets against the IGC50, LC50, LC50DM and LD50 datasets. For IGC50 and LD50 SENN reveals the best performance: RMSE of 0.415 ± 0.001, MAE of 0.301 ± 0.002, and RMSE of 0.557 ± 0.002, MAE of 0.431 ± 0.003, respectively. All comparisons besides LC50 and LC50DM are statistically significant.

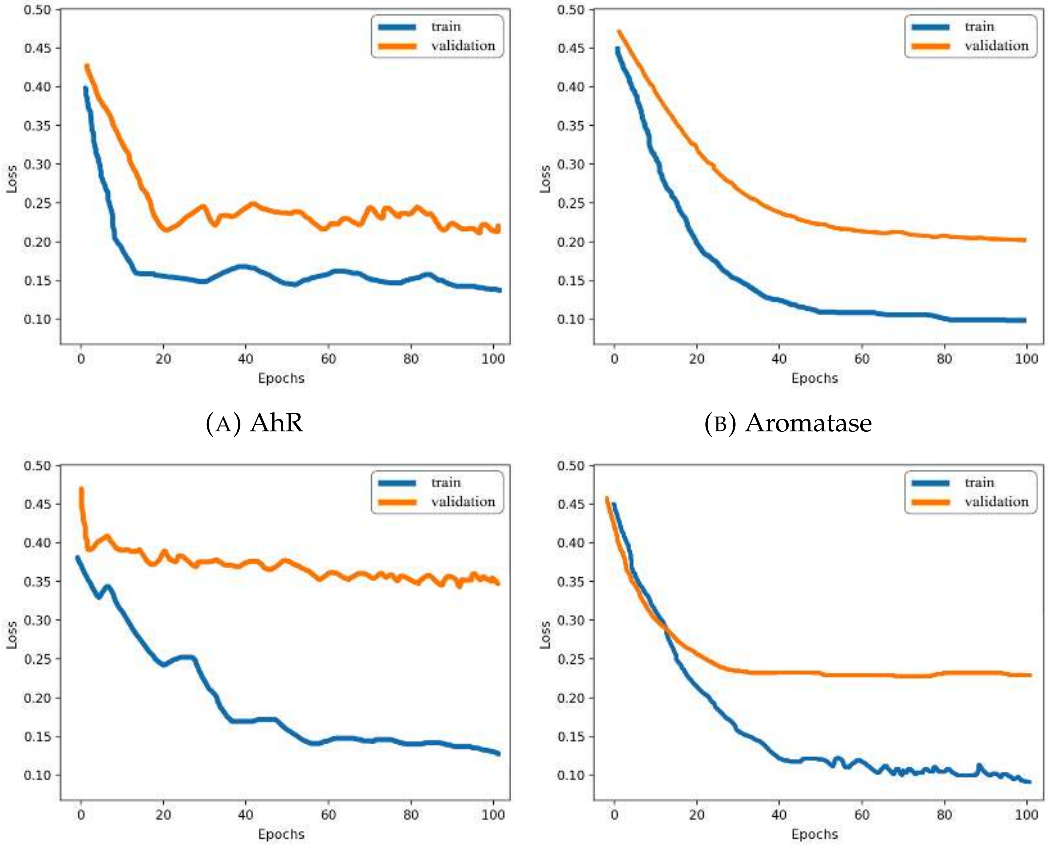


(C) LC50DM



(D) LD50

FIGURE 4.3: The RMSE and MAE scores on the test sets against the IGC50, LC50, LC50DM and LD50 datasets. For IGC50 and LD50 SENN reveals the best performance: RMSE of 0.415 ± 0.001, MAE of 0.301 ± 0.002, and RMSE of 0.557 ± 0.002, MAE of 0.431 ± 0.003, respectively.



(A) AhR

(B) Aromatase

(C) ATAD5

(D) p53

FIGURE 4.4: SENN training and validation losses.

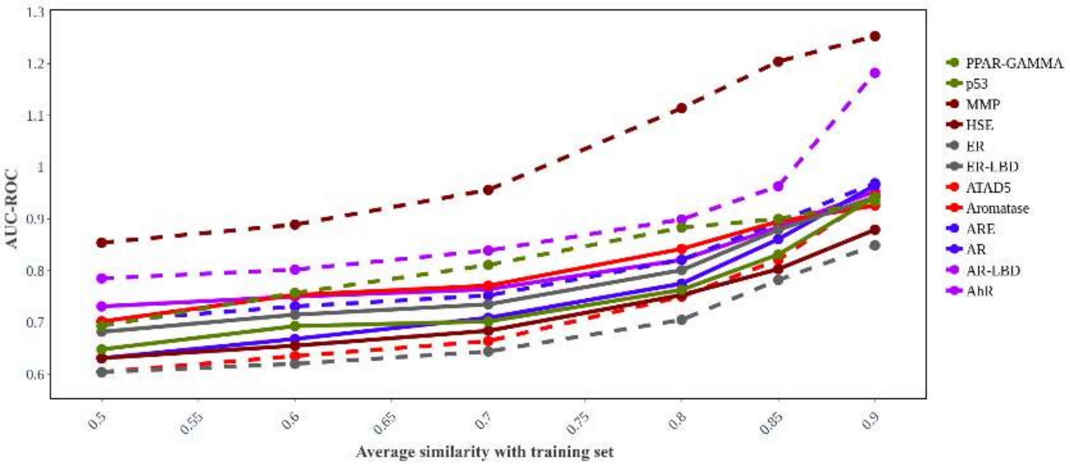


FIGURE 4.5: AUC-ROC vs similarity. One may notice a correlation between our model performance and average molecules similarity. In other words, SENN's performance decreases when the average similarity decreases between the molecules, for which the compound score is predicted, and the rest of the training set molecules.

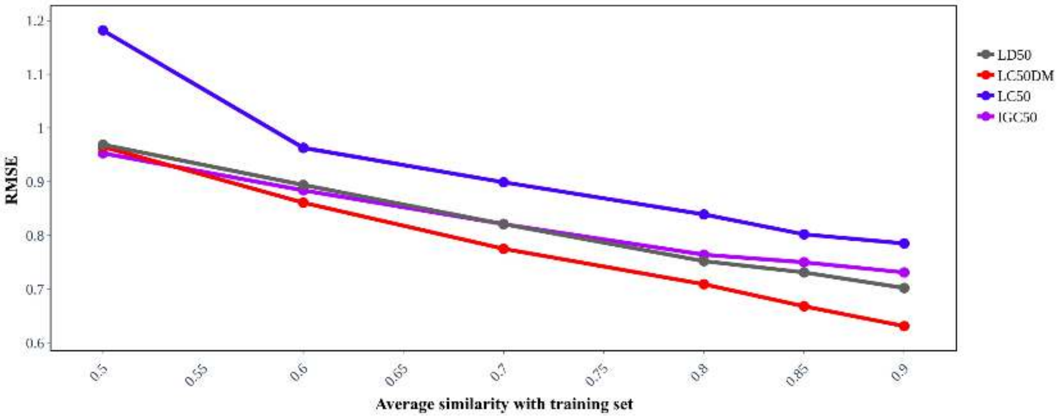


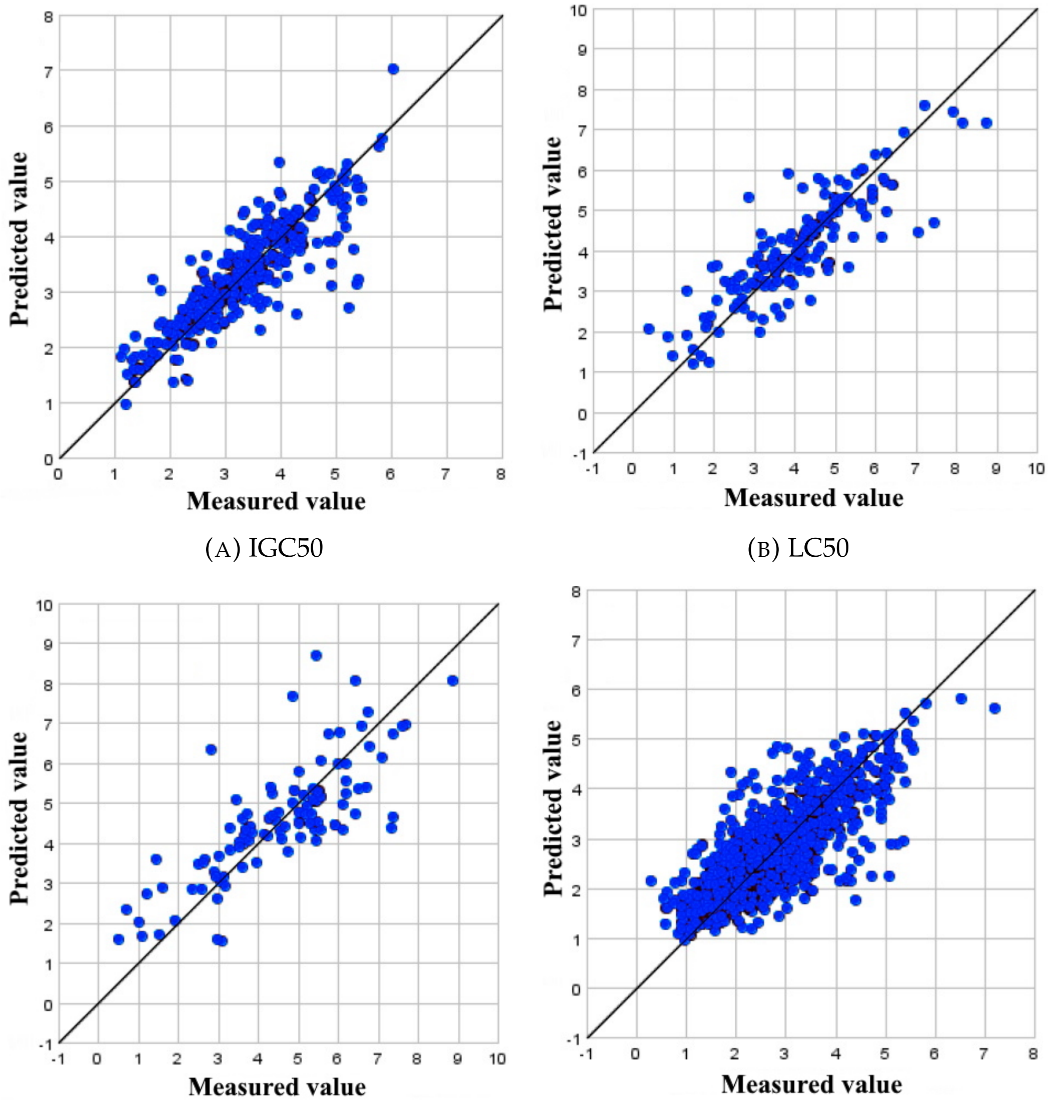
FIGURE 4.6: RMSE vs similarity. One may notice a correlation between our model error and average molecules similarity. In other words, SENN's performance decreases when the average similarity decreases between the molecules, for which the compound score is predicted, and the rest of the training set molecules.

4.4.6 Analysis of distribution of experimental and predicted value

Table 4.5 and Figure 4.7 show the distribution of experimental and predicted values on test subsets for the IGC50, LC50, LC50DM, and LD50 dataset. Our analysis indicate that both TopTox and SENN come up with R^2 values greater than 0.7 except for the LC50DM which has $R^2 = 0.641$ and $R^=$0.628 in case of TopTox and SENN, respectively. The lower correlation value for the LC50DM dataset is probably due to the fact that we the number of compounds contained in this dataset is very limited and we used a mixture of LC50 and LC50DM dataset to train the model. IGC50, LC50 and LD50 have the R^2 value of 0.764 , 0.731 0.712 for SENN, and 0.748, 0.745, 0.701 for TopTox. It denotes they have almost equal capability in prediction of the real derivative data through the nonlinear regression analysis.

TABLE 4.5: RMSE for validation sets (lower is better).

Dataset	TopTox	SENN (ours)
IGC50	0.748 ±0.009	0.764 ±0.011
LC50	0.745 ±0.01	**0.731** ±0.009
LC50DM	0.641 ±0.013	**0.628** ±0.010
LD50	0.701 ±0.008	0.712 ±0.009



(A) IGC50

(B) LC50

(C) LC50DM

(D) LD50

FIGURE 4.7: Confusion matrix. The R^2 scores of both TopTox and SENN are greater than 0.7 except for the LC50DM which has $R^2 = 0.641$ and $R^{=}$0.628 in case of TopTox and SENN, respectively.

4.4.7 Interpretability of SENN

We now turn our attention to the interpretability of SENN. Therefore, four toxic chemical compounds are randomly sampled from the Tox21 Challenge dataset. Then we investigate chemical substructures which are relevant for classification into toxic molecules (Figure 4.8). Herein, the Integrated Gradients (IG) feature attribution technique was employed [121].

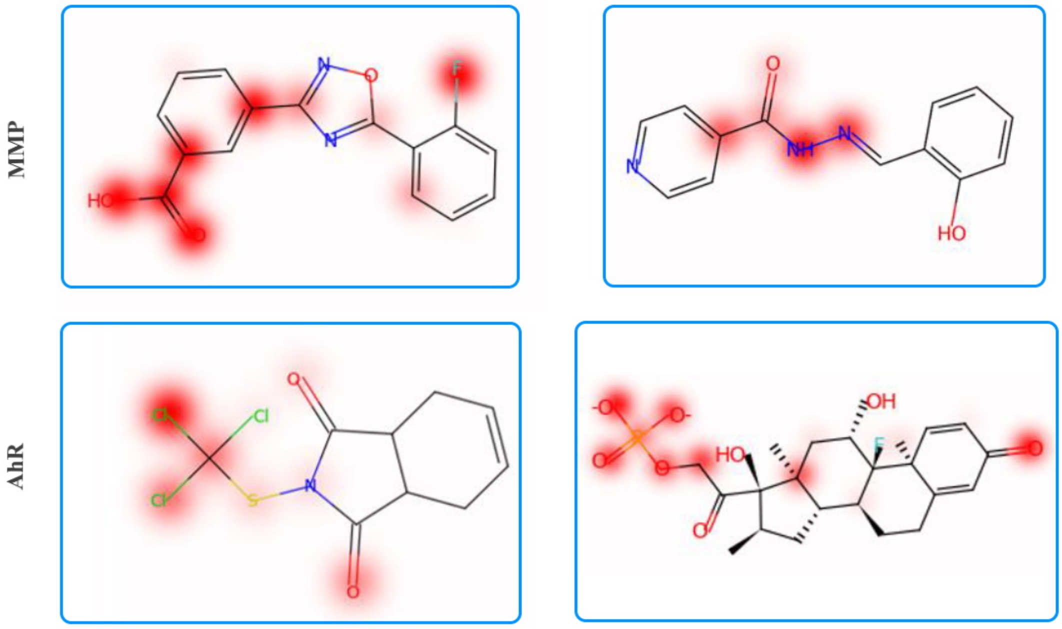


FIGURE 4.8: Heatmaps for the atoms for four randomly selected samples. Dark red colour indicates that these atoms are responsible for a positive classification. SENN correctly identifies the chemical substructures (toxicophores) that may cause toxicity such as hydrazones.

IG makes it possible to get feature attributions relative to an uninformative baseline. In other words, it evaluates the extent to which each input feature contributes to the output by computing an average partial derivative of each feature as the input varies from the baseline to its final value.

In Figure 4.8, we demonstrate attributions produced by Integrated Gradients on a few examples from the dataset. According to the explanations, certain chemical groups obtain more positive attributions (marked in red) by IG, and such an observation matches the ground truth. This means that SENN

TABLE 4.6: Ablation study on the effect of removal of the global physicochemical features. In general, the variant which only learns the representation using fragments reveals a significant drop of performance for all validation datasets.

Dataset	SENN	SENN without extra physicochemical features
AR	0.802	0.790
Aromatase	0.821	0.809
ER-LBD	0.813	0.802
PPAR-gamma	0.821	0.815
AhR	0.902	0.895
AR-LBD	0.839	0.811
ER	0.710	0.697
ARE	0.790	0.782
p53	0.819	0.808
MMP	0.906	0.894
HSE	0.765	0.753
ATAD5	0.835	0.829

is able to identify a strong association. As the Figure 4.8 shows, it can be noticed that our model is able to correctly identify the chemical substructures (toxicophores) that may cause toxicity, such as hydrazones.

4.4.8 Ablation study

We also conduct ablation study to investigate the validity of key components of SENN. As shown in Table 4.6, decoupling physicochemical features in the learning framework yields a significant drop of performance for all datasets. For instance, in the case of AR-LBD dataset, SENN returns AUC-ROC of 0.839 but its reduced version (without extra physicochemical attributes) achieves AUC-ROC of 0.811. This evidence supports our claim that injecting chemical domain knowledge into architecture improves model behaviour.

In the next experiment, we compare the performance of SENN with the method introduced in Chapter 3, i.e. HybNN. Only for ER-LBD HybNN beats SENN (see 4.9). It indicates that fragment-based approach employed in SENN helps to capture the molecular information.

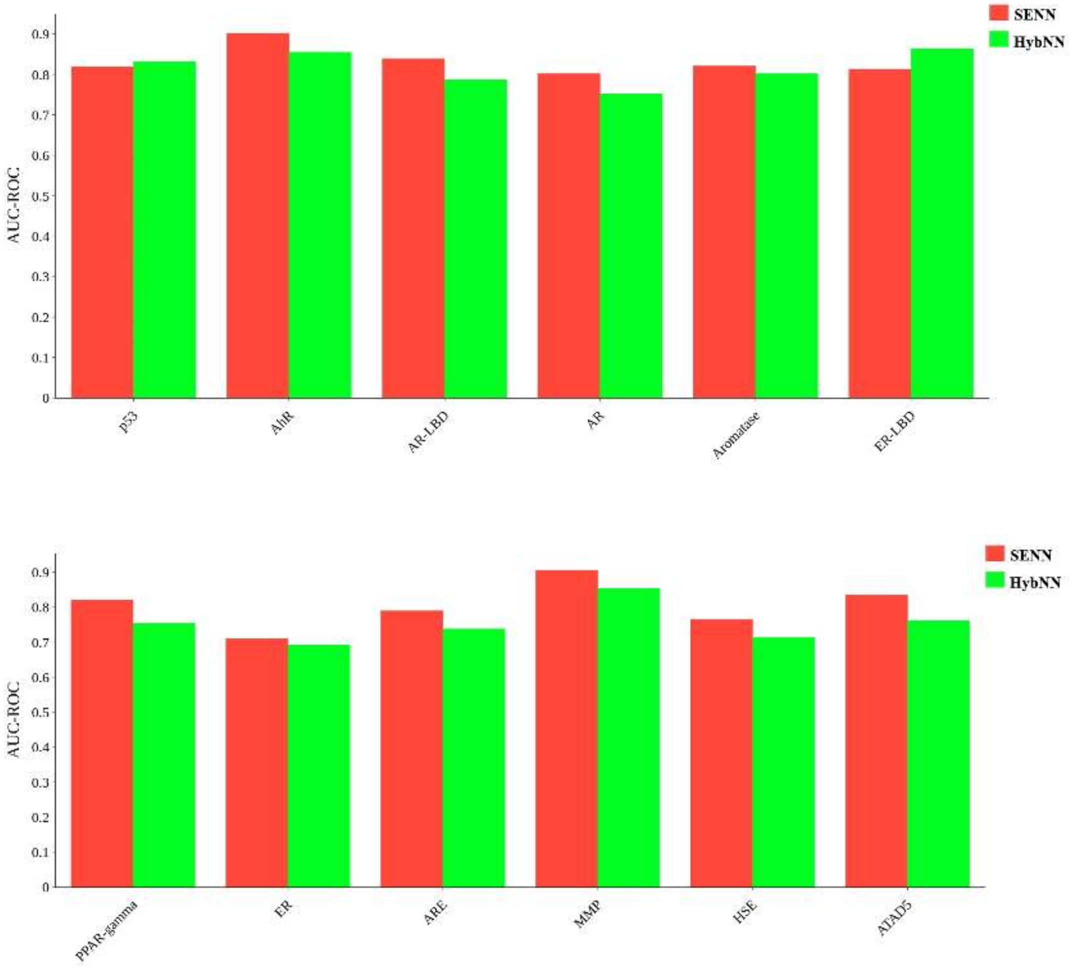


FIGURE 4.9: Comparing SENN and HybNN (presented in Chapter 3). Results are the mean values for the test set after five independent trials of training. Only for ER-LBD HybNN beats SENN which suggests that fragment-based approach employed in SENN helps to capture the molecular information

4.5 Conclusion

In this work, we proposed Subgraph Encoded Neural Network (SENN) to enable more accurate predictions of molecular properties. In SENN, we employ a subgraph embedding component that is fed into graph convolutional networks to improve the learning process. We further boost the expressive power of SENN by proposing a set of chemical attributes that enhance information associated with molecular physicochemical features. Extensive experiments demonstrate that SENN achieves remarkably superior performance over the state-of-the-art models on various publicly available regression and classification benchmarks.

Chapter 5

Deep Representation Learning of Graphs and Sequences

This chapter presents our third contribution. We apply the representation learning framework to the task of predicting drug-target interactions in heterogeneous networks. Here, a new model, namely **T**riplet **E**ncoded **N**eural **N**etwork (TENN), comprised of three main components is presented. One of the components learns low-dimension vector representations for target that is a protein sequence, by extracting relevant information from the characters string. In turn, the other two units learn two representations of chemical compounds. This is done by well-designed approaches based on a graph attention network (GAT) and a continuous bag of molecular words. Our experimental results demonstrate the ability of our method to achieve competitive prediction performance against existing state-of-the-art models (KronRLS [117], DeepCPI [149], SimBoost [71], DeepDTA [115]) over biologically plausible drug-target interaction datasets.

5.1 Introduction

As we have seen in Chapters 2, 3 and 4, challenges such as detection of bioactive chemical compounds, as well as molecular properties prediction including toxicity, are all examples of drug discovery tasks. Nevertheless, among computational approaches to drug development, the scientific community has already made tremendous progress in the identification of drug-target interactions (DTI) [176]. DTI is significant, especially for finding effective and safe treatments. It is also worth to mention that majority of the existing DTI works have formulated the DTI prediction task as a binary classification.

In the pharmaceutical sciences, a drug target is a chemical compound that is capable of binding to drugs and producing effects on cells. Proteins are considered as obvious molecular targets [49]. Of course, in the literature, there is a great variety of in silico proposals on DTI prediction. Studies have examined several ligand-based [89] and docking-based methods [46]. For instance, KronRLS [117] uses the Kronecker Regularized approach for minimizing a cost function and employs the similarity matrices for drugs and targets to obtain the results. Another well-known approach is SimBoost [71] that is based on the assumption that molecules with similar structures are more likely to reveal similar effects. In their work, He et al. made attempts to predict binding affinity scores with a gradient boosting machine by using feature engineering to represent drug-target interactions. However, traditional machine learning methods are not commonly used at present to predict DTIs, as researchers have found a few relevant drawbacks. Firstly, these approaches usually need a large number of known binding data. In consequence, the prediction results are not satisfactory when one works with a small amount of known data. Secondly, the performance is much worse if the three-dimensional structures of the target protein are not available.

To address the aforementioned shortcomings, a novel DTI prediction methodology, called **T**riplet **E**ncoded **N**eural **N**etwork (TENN) is introduced. TENN aims to identify the drug-target interactions by exploiting the existing topological structure of drug molecules, along with modeling spatio-sequential information. To sum up, here are three major contributions:

- We propose a learning-based method for drug-target interaction prediction that contains three components.
- We show that if one extracts the global information of protein sequences and drug compounds, it leads to not only an improvement in the efficiency of DTI, but enables to detect more complex interactions.
- Based on our DTI prediction task, the results indicate that TENN is better than the other four state-of-the-art approaches.
- The core advantage of TENN is the ability to handle the low-dimensional feature vectors and predict the probability of interaction between each pair of drugs and proteins.
- Although our research on TENN focuses on the application to the problems in chemistry, the proposed methodology is universal and could be employed to model various interactions in the world.

5.2 Related Work

Lately, the availability of pharmacological databases coupled with advances in computational resources have engendered the growth of deep learning-based methods dealing with drug-target interaction prediction. Take, for instance DeepDTA, in which the authors model protein sequences and compound 1D representations with convolutional neural networks (CNNs) [115].

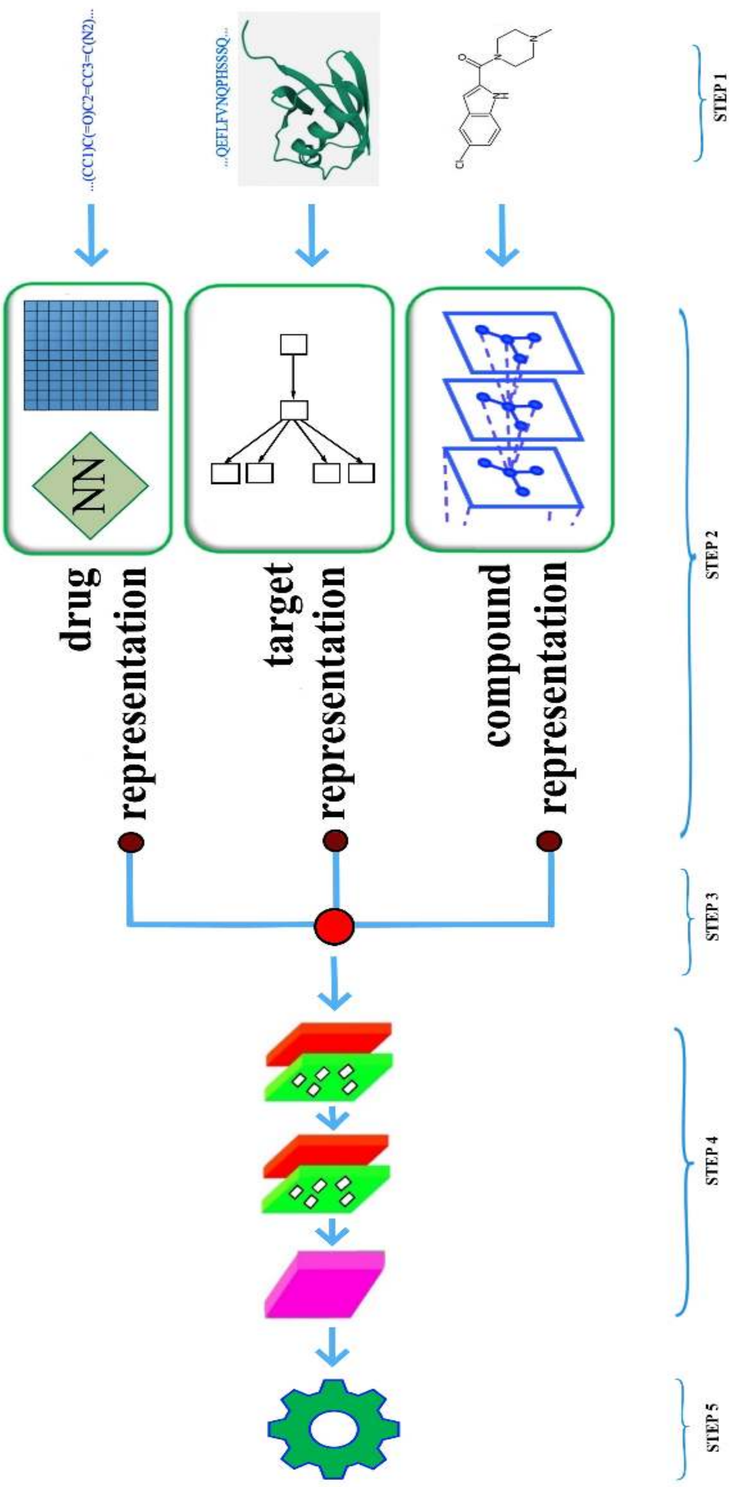


FIGURE 5.1: TENN architecture. The main novelty is the heterogeneous network that integrates a variety of drug and protein related information sources by employing three separate units (see Figure 5.2, Figure 5.4 and Algorithm 1). Each unit learns a representation. The three representations are then combined and fed into a set of linear layers with dropout to make a final prediction.

This technique has revealed significant performance on a kinase family bioassay dataset. What is more, according to Öztürk, the more data, the hidden information is captured better. Nevertheless, DeepDTA transforms drug compounds and protein sequences into a corresponding string representation that is not necessarily an effective way to characterize molecules. Another example is DeepCPI [149]. To be more precise, Tsubaki et al. propose a deep learning-based method for modeling DTI by using latent semantic analysis and natural language processing techniques to learn feature embeddings. They used two sets of compound-protein interactions datasets for human and C.elegans. The experiments confirm that the methodology based on convolution and graphs can achieve competitive or higher performance than the baseline approaches. However, utilizing a conventional convolution may lead to the loss of protein structure information. In addition, although several aspects have been addressed, Tsubaki does not also exploit enough the topological structure of molecules. On the other hand, Pahikkala et al. propose the Kronecker Regularized Least Squares method that employs 2D based compound similarity-based representations of the drugs and Smith-Waterman similarity representation of the targets [117]. Furthermore, the authors have pointed out four factors that either alone or together with the other factors bring about considerable differences in the prediction results. Despite the promising predictive performance of Pahikkala's model, it depends on the nature of feature engineering employed.

Even though in recent years numerous applications of these methodologies have been seen, they still may be improved. One may notice that many techniques allow to capture invariant local patterns but do not take into consideration the long-term dependencies, including relationship between objects that are far apart each other. Moreover, many methods often fail to predict a potential interaction for given new targets. Therefore, more significant efforts have to be made to explore the extent to which the novel approaches promote solutions that contain diverse exploration strategies of chemical compounds.

5.3 Methodology

We propose a learning-based method, namely TENN, to predict drug-target interactions. The workflow of TENN is presented in Figure 5.1. First of all, the heterogeneous network was constructed by integrating a variety of molecule- and protein-related information sources in a form of three components. Since we use two different approaches to extract meaningful features from that chemical compound, we refer to it as a (hypothetical) drug and just a compound. To be more precise, TENN operates on tokens in both drug and protein sequences and on a graph of a molecule. Each component is designed to calculate a distributed representation of the input. Therefore, component no. 1 returns $out_{drug} \in \mathbb{R}^F$, component no. 2 yields $out_{target} \in \mathbb{R}^F$ and the output of component no. 3 is $out_{compound} \in \mathbb{R}^F$. In the second step, the F-dimensional feature vectors of drugs, molecules, and proteins are combined (see step 5 in Figure 5.1) and outputs into a single feature vector $y_{cout} \in \mathbb{R}^{3F}$. Finally, three linear layers with a dropout layer after the first two are added (see step 4 in Figure 5.1). In the last part, we predict the association between a chemical compound (possible future drug) and a protein.

5.3.1 Component 1: drug representation

We break each ligand SMILES into a 'sentence' comprising biological subwords (chemical words) in an overlapping manner. To be more precise, k-mers are extracted, where $k = 4$ [116]. To do this, we traverse a window of length k over the SMILES string of a ligand and extract all overlapping SMILES substrings of length k. Consequently, each k-mer is seen as a chemical word $\{s_1, s_2, \ldots, s_L\}$, where L is the number of possible k-mers for the selected molecule. For example, let us assume the SMILES string for Ampicillin is as follows:

“CC1(C(N2C(S1)C(C2=O)NC(=O)C(C3=CC=CC=C3)N)C(=O)O)C”. When one splits it, the total number of possible 4-mers is 47.

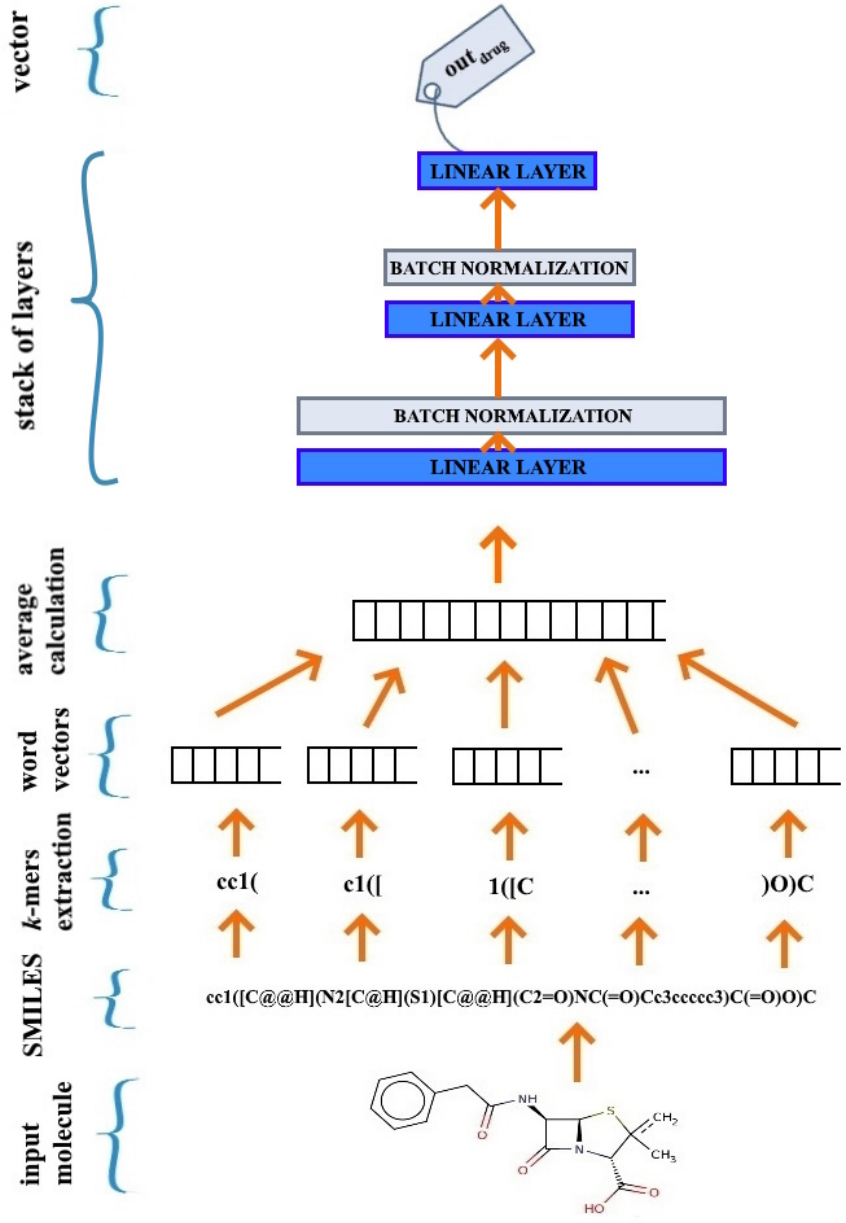


FIGURE 5.2: TENN: component 1. Each ligand SMILES is broken into a ’sentence’ comprising biological subwords (chemical words) in an overlapping manner. Then, the subwords embeddings are learnt. The obtained vectors are averaged and fed into two linear layers.

Then, the Word2Vec [109] algorithm is employed, continuous-bag-of-words (CBOW) approach, to learn embeddings for these chemical words by training on approximately 1.7M canonical SMILES strings that are collected from the ChEMBL [108] database. Finally, the obtained vectors $\{sv_1, sv_2, \ldots, sv_L\}$, where $sv_i \in \mathbb{R}^{dim}$ are averaged and the computed vector average can be represented as $\{w_1, w_2, \ldots, w_m\}$, where $w_i = \frac{1}{L}\sum_{j=1}^{L} sv_{ji}$, and m denotes the fixed-length of the feature vector (average calculation step in Figure 5.2). If $L < m$, zero is appended at the end.

Then, the computed feature vector is fed into a neural network that is composed of two linear layers with batch normalization (as shown in Figure 5.2). The output is drug representation, $out_{drug} \in \mathbb{R}^F$.

5.3.2 Component 2: target representation

Component 2 comprises two pathways which are called unit A and unit B. In the end, component 2 outputs the protein representation as the feature vector $out_{target} \in \mathbb{R}^F$.

Protein representation A single protein is usually represented as a sequence of 20-letter alphabets which are associated with a particular amino acid. Therefore, here, a protein is a sequence of amino acids.

Unit A: Our goal is to transform each protein sequence into a vector representation. Therefore, to learn feature representations, Word2vec [109] is adopted. According to our methodology, each amino acid is treated as a word and a protein sequence is seen as a sentence. In consequence, 557000 protein sequences from the Swiss-Prot database [6] are fed into the Word2vec model to obtain the representation of different biophysical and biochemical properties that are supposed to be hidden in protein sequences. It is assumed that the output of the Word2vec model are amino acid feature vectors. As a result, each protein sequence in our approach is represented as a concatenation of amino acid vectors (see Figure 5.3).

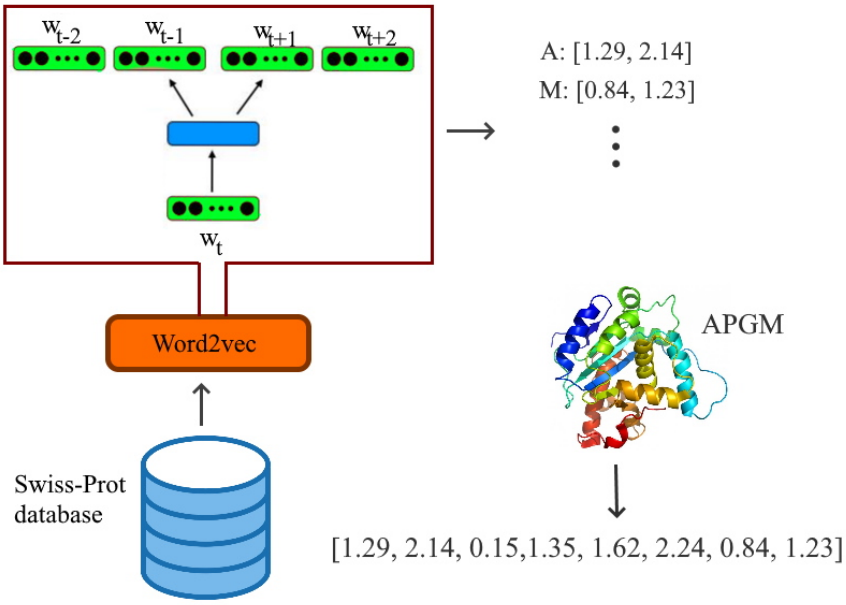


FIGURE 5.3: TENN: component 2 (unit A). Each amino acid is seen as a word and a protein sequence is treated as a sentence. Therefore, the protein sequences are fed into the Word2vec model to obtain the representation. Finally, each protein sequence is represented as a concatenation of amino acid vectors.

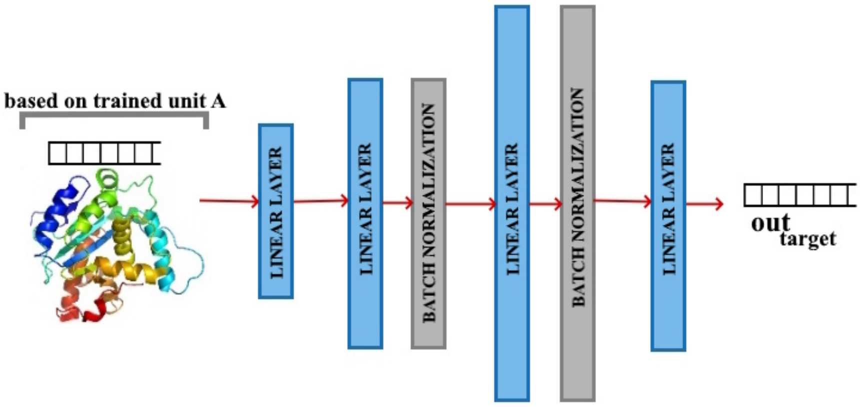


FIGURE 5.4: TENN: component 2 (unit B). First, an embedding vector with a fixed size from unit A is used to describe a protein. Second, the unit B delegates execution of linear layer, a stack of linear layer with batch normalization and last linear layer. Then the final resprentation out_{target} is returned.

Unit B: The model from unit A is fetched to create word embeddings for proteins used to train unit B. These vectors are employed as input for unit B.

In turn, unit B consists of a linear layer, two pairs of linear layer with batch normalization and an extra linear layer. The output stores the final vector representation, $out_{target} \in \mathbb{R}^F$ (see Figure 5.4).

5.3.3 Component 3: compound representation

According to our method, the input to this component is a graph. Therefore, the RDKit Open Source software [97] was employed to transform the SMILES string of a chemical compound into a molecular graph $\mathcal{G} = (\mathcal{V}; \mathcal{E})$, where the vertex $v_i \in \mathcal{V}$ represents the ith atom, and $e_{ij} \in \mathcal{E}$ corresponds to the chemical bond between the atom v_ith and the atom v_jth. After the transformation, the graph $\mathcal{G}$ is passed through by a graph attention network (GAT) [154]. However, in order to use GAT with molecular graph, one has to encode atoms to a d-dimensional vector. Then, we aggregate the information from the k-paths subgraph 4.3.2 for each atom in the molecular graph. The pseudo-code of this methodology is shown in Algorithm 1.

The approach works as follows. Firstly, we compute an initial vector for each atom (line no. 4). Then, the Weisfeiler-Lehman-based (C-WL) algorithm is employed [80]. In the initial version, C-WL calculates embedding vectors for each combination of an atom and its neighbors. We employ this concept to provide a vector representation for the atoms. However, we extend C-WL by incorporating the k-path parameter. In consequence, depending on the value of k (line no. 5) more or less neighbors are considered. Then, these two embedding vectors are combined by concatenation (line no. 6). The obtained node embedding $V' = \{v'_1, v'_2, \ldots, v'_{|\mathcal{V}|}\}$, where $v'_i \in \mathbb{R}^{d_3}$, are subsequently fed into a graph attention network (GAT) module. In the end, we are able to gather atom vectors $V'' = \{v''_1, v''_2, \ldots, v''_{|\mathcal{V}|}\}$, where $v''_i \in \mathbb{R}^F$ to obtain an F-dimensional representation of the molecule by summing up the node vectors (line no. 9).

Algorithm 1 Graph Attention Network (GAT) for the compound representation

```
Input: Molecule as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, length $k$, $\tilde{N}_i$ - the neighboring nodes of $i$th atom, $W \in \mathbb{R}^{F \times d_3}$, $\alpha_{ij}$ - attention coefficient between the $i$th atom and the $j$th atom
Initialize: $out_{compound} \leftarrow [0, \ldots, 0]$.
for each atom $a$ in molecule do
    $a_{nr} \leftarrow \mathcal{G}.get\_node\_representation(a) \in \mathbb{R}^{d_1}$
    $a_{kr} \leftarrow \mathcal{G}.get\_k\_distance\_nodes\_representation(a, k) \in \mathbb{R}^{d_2}$
    $v'_a \leftarrow Concatenation(a_{nr}, a_{kr}) \in \mathbb{R}^{d_3}$
for each atom $a$ in molecule do
    $v''_a = \sigma(\sum_{j \in \tilde{N}_a} \alpha_{aj} W v'_j)$
Output: $out_{compound} \leftarrow \sum a; \forall a \in \mathcal{V}''$
```

5.4 Experiments and Results

5.4.1 Data collection

In order to evaluate the proposed methodology, TENN was tested on the drug-target interactions prediction task. Specifically, the data were downloaded from BindingDB [103] database that contains experimentally determined binding affinities on the interactions of target proteins with small, drug-like molecules. The database stores 1756093 binding data for 7371 protein targets and 780240 small chemical compounds. To clean up the data, a few steps were taken. Firstly, to ensure relevance of the properties, we excluded inorganic compounds and protein targets with sequence identity more than 75%. Secondly, we removed interactions samples where the IC50 value was missing or equated more than 300 nM. Our goal was to leave compounds that are most likely of being able to modulate a target with a small-molecule drug. Note that IC50 denotes the effectiveness of a drug in inhibiting the growth of a specific enzyme. The less the value, the more effective the drug may be. As a result, we obtained 36014 small molecular drugs and 2,099 targets which gives more than 75 million DTI pairs. Please note that 83676 pairs are known as positive DTIs, but the remaining are undetermined. In

TABLE 5.1: Dataset details.

Dataset name	#Targets	#Drugs	#Interactions
dataset1	3839	6068	15434
dataset2	3839	6068	3348
dataset3	3839	537	1735
dataset4	160	6068	264
dataset5	160	537	37

consequence, we randomly selected 83676 drug-target pairs from the second set of pairs and treated them as negative data.

Furthermore, we also obtained known DTIs from DrugBank [164]. The date of May 15, 2016 was the factor that separated the data. Thus, we split the data into two groups, i.e. 'new' and 'old'. The new data is the data that appeared after the chosen date, and the remaining data is assumed as old. Finally, the data was grouped into five datasets labeled as positive as listed in Table 5.1: (dataset no. 1) comprised of old drugs, old targets and their old interaction pairs; (dataset no. 2) comprised of old drugs, old targets and their new interaction pairs; (dataset no. 3) comprised of new drugs, old targets and their interaction pairs; (dataset no. 4) comprised of old drugs, new targets and their interaction pairs; (dataset no. 5) comprised of new drugs, new targets and their interaction pairs. Dataset no. 1 was selected for training, and the remaining four datasets were employed for evaluation of TENN. In turn, in order to select the negatively labeled dataset, we generally followed the same steps as in the case of the BindingDB.

5.4.2 Evaluation

We compare TENN with state-of-the-art models such as KronRLS [117], DeepCPI [149], SimBoost [71], and DeepDTA [115] in four datasets mentioned in Subsection 5.4.1. Figure 5.5 reports the average AUC-ROC (AUC), Precision (Pre), Recall (Rec), Accuracy (Acc), and F1 scores over 3 runs with different random seeds on four datasets, as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

$$Pre = \frac{TP}{TP + FP} \tag{5.2}$$

$$Rec = \frac{TP}{TP + FN} \tag{5.3}$$

$$F1 = 2\frac{Pre * Rec}{(Pre + Rec)} \tag{5.4}$$

where where TP is true positive, TN is true negative, FP is false positive, FN is false negative.

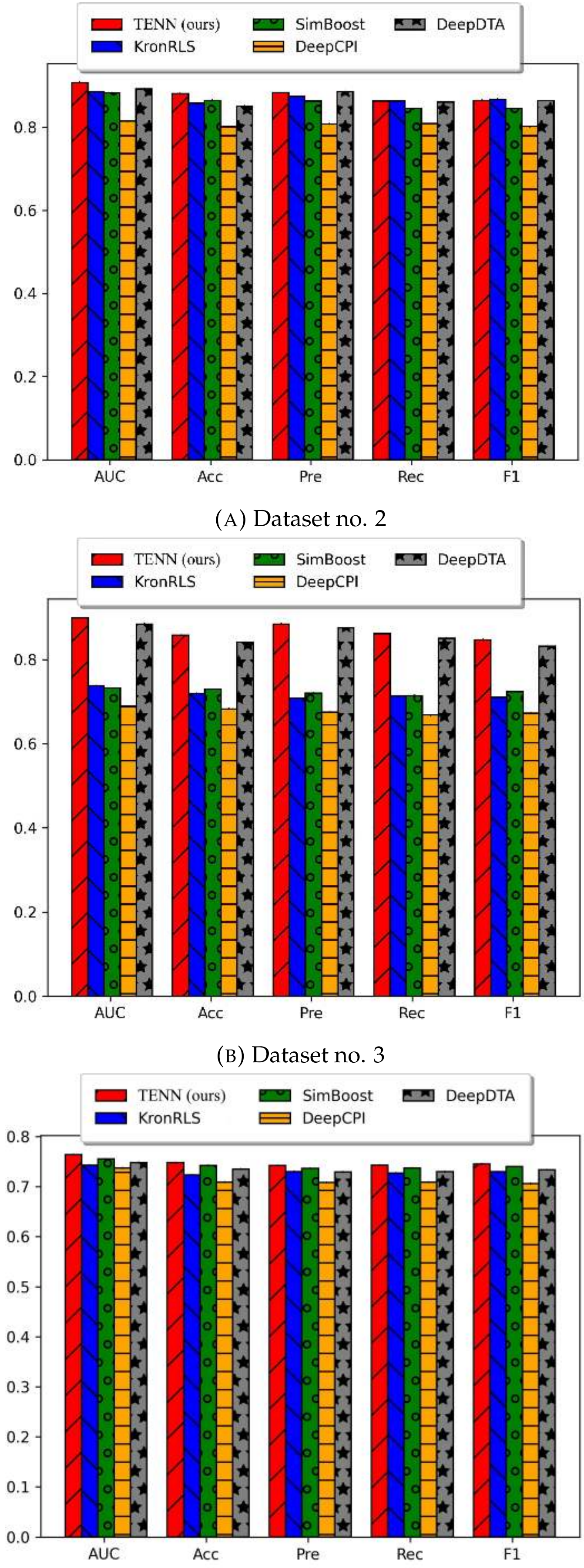
Moreover, we evaluate statistical significance using a one-sided Wilcoxon signed-rank test. If a p-value is less than 0.05, it indicates the outcomes are statistically significant.

5.4.3 Performance comparison with other models

Figure 5.5 shows the comparison of our method and the other machine learning techniques in terms of different measures such as AUC-ROC (AUC), Accuracy (Acc), Precision (Pre), Recall (Rec) and F1. One can find that our method outperforms the competing methods. In case of dataset no. 2, TENN achieves AUC-ROC, Accuracy, Precision, Recall and F1 of 0.908 ± 0.004, 0.885 ± 0.004, 0.883 ± 0.004, 0.815 ± 0.004 and 0.892 ± 0.007, respectively. The second highest AUC-ROC, Accuracy, F1 result has SimBoost (0.884 ± 0.004, 0.874 ± 0.005, 0.885 ± 0.003). As far as dataset no. 3 is considered, our model also reveals competitive performance with scores 0.899 ± 0.004 (AUC-ROC), 0.738 ± 0.004 (Accuracy), 0.732 ± 0.003 (Precision), 0.689 ± 0.004 (Recall) and 0.884 ± 0.007 (F1). For instance, SimBoost, which is the second-best approach, achieves AUC-ROC = 0.884 ± 0.004 and F1 = 0.875 ± 0.004. Moreover, TENN beats other methods for dataset no. 4, where it has 0.764 ± 0.005 (AUC-ROC), 0.743 ± 0.004 (Accuracy), 0.755 ± 0.003 (Precision), 0.737 ± 0.004 (Recall), 0.748 ± 0.005 (F1). To compare, KronRLS that can be treated as the second-best method, shows AUC-ROC of 0.748 ± 0.004, and F1 of 0.735 ± 0.004. Last but not least, is dataset no. 5, where TENN achieves AUC-ROC, Accuracy, Precision, Recall and F1 of 0.797 ± 0.005, 0.667 ± 0.004, 0.779 ± 0.003, 0.725 ± 0.004, and 0.744 ± 0.005, respectively. Here, DeepCPI and reveals the second-best performance with AUC-ROC of 0.789 ± 0.003, and F1 of 0.727 ± 0.004. In addition, all comparisons besides dataset no. 4 are statistically significant.

To sum up, our approach performs the best, and we find that there are at least two reasons behind it. First of all, in contrast to the other methods, TENN does not rely heavily on hand-crafted features. Secondly, we consider both topological structures and local chemical context information.

We also used ROC curves to measure the comprehensive index between the false positive rate and the true positive rate continuous variable. The goal was to investigate why DeepDTA performs significantly worse than our approach. Figure 5.6 illustrates the ROC curve of the TENN predictor and DeepDTA predictor. As it was mentioned earlier, our averaged AUC-ROC reaches 0.899, while DeepDTA achieves 0.847. We interpret the worse performance of DeepDTA is caused by the fact that the method is optimized for a densely constructed dataset with specific protein class (is optimized for KIBA [144] and Davis [37] dataset), while the training datasets used in this study cover various protein classes.

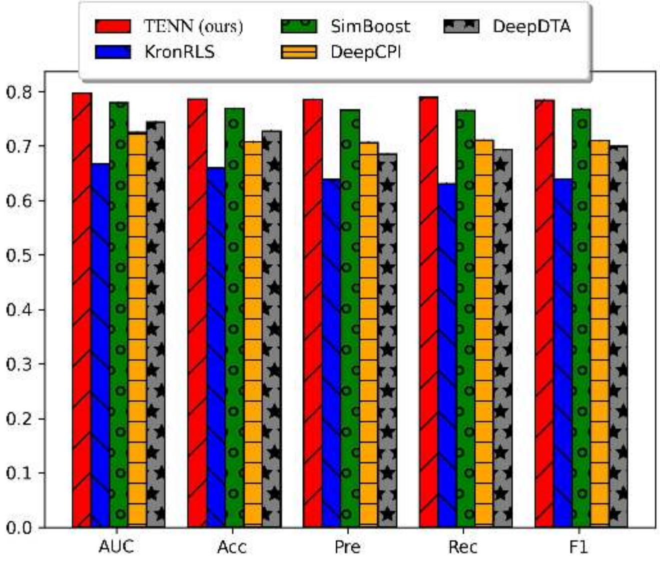


(A) Dataset no. 2

(B) Dataset no. 3

(C) Dataset no. 4

FIGURE 5.5: The performance of TENN over all datasets. It may be observed that TENN outperforms the other approaches. A ll comparisons besides dataset no. 4 are statistically significant.



(D) Dataset no. 5

FIGURE 5.5: The performance of TENN over all datasets. It may be observed that TENN beats the other approaches.

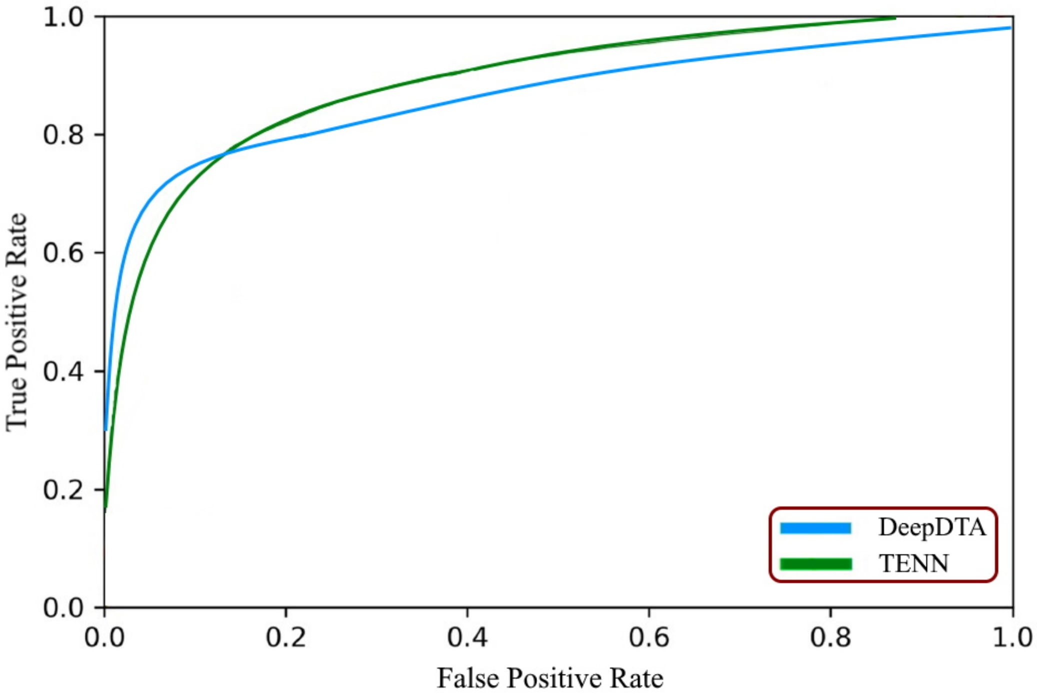
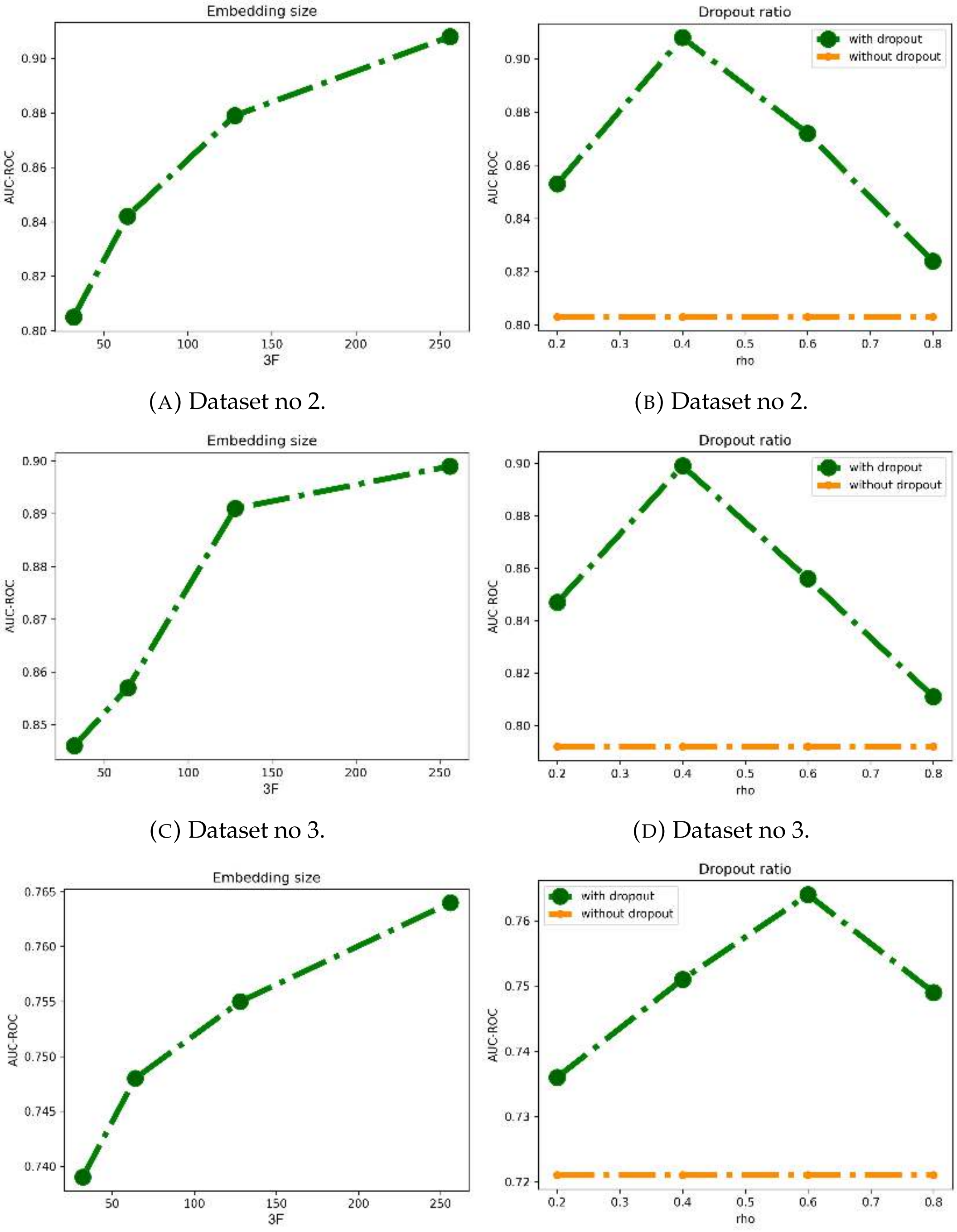


FIGURE 5.6: The ROC curve of TENN and DeepDTA. Our averaged AUC-ROC reaches 0.899, while DeepDTA achieves 0.847. TENN performs better since it is optimized for various data sources.



(A) Dataset no 2.

(B) Dataset no 2.

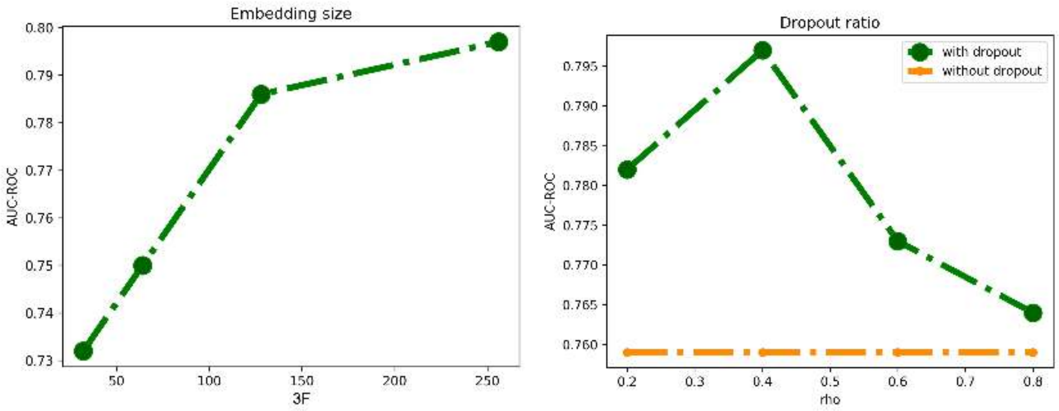
(C) Dataset no 3.

(D) Dataset no 3.

(E) Dataset no 4.

(F) Dataset no 4.

FIGURE 5.7: Figure 5.7a, 5.7c, 5.7e and 5.7g show how the embedding size $3F$ is related to the length of final representation affects the predictive performance of TENN. Our model benefits from a large embedding size. Figure 5.7b, 5.7d, 5.7f, and 5.7h investigate the impact of dropout regularization. The dropout substantially increases the overall performance of TENN.



(G) Dataset no 5.

(H) Dataset no 5.

FIGURE 5.7: Figure 5.7a, 5.7c, 5.7e and 5.7g show how the embedding size $3F$ related to the length of final representation affects the predictive performance of TENN. Our model benefits from a large embedding size. Figure 5.7b, 5.7d, 5.7f, and 5.7h investigate the impact of dropout regularization. The dropout substantially increases the overall performance of TENN.

5.4.4 Ablation study

We perform some ablation studies to investigate the impact of an embedding size and a dropout ratio in TENN. Figure 5.7 shows the performance of our default method and its variants.

- Embedding size of TENN. The embedding size $3F$ affects the representation ability of TENN. We vary $3F$ within {33, 63, 129, 255}. As shown in Figure 5.7a, 5.7c, 5.7e, and 5.7g TENN benefits from a large embedding size. The results on other scenarios have similar trends.

- Dropout regularization in TENN. Figure 5.7b, 5.7d, 5.7f, and 5.7h show the performance of TENN vs. a dropout ratio. Our results show that dropout offers better performance. In fact, depending on the dataset, the dropout rate must be tuned. For instance, using the dropout ratio $\rho \approx 0.4$, TENN achieves a favorable AUC-ROC for the dataset no. 3.

5.5 Conclusion

In this chapter, we introduced a novel methodology, Triplet Encoded Neural Network (TENN), for drug-target interactions prediction. The well-designed model employs three units to learn the representations of the drug, target and chemical compound level by level, and then make prediction with overall interaction representation. The experimental results on publicly available datasets demonstrated the competence of our method.

Chapter 6

Afterword

A tree that is unbending, is easily broken.

—*Lao Tzu*

A person with a new idea is a crank until the idea succeeds.

—*Mark Twain*

6.1 Conclusion

In this Thesis, we have presented various combinations of representation learning models to enhance drug discovery.

First, we tackled the problem of drug bioactivity prediction (Chapter 3). By a well-designed architecture that integrates two different concepts, such as a stack of convolutional layers, and an RNN based on bidirectional gated recurrent unit (BiGRU), we were able to obtain a final representation. Also, we employed a collection of molecular features that aim to capture the structure-property relationships. This allowed us to build a classifier to detect active and inactive chemical compounds.

In Chapter 4, we identified various shortcomings of existing approaches for toxicity prediction in which the key limiting factor, among others, is the requirement for large datasets from which to train complex models. By exploiting a graph structure the learning process was improved. Furthermore, we showed that incorporating a set of global molecular features such as a molecular weight or a number of rotatable bond increases discriminative capability of our model for classification and regression tasks.

Lastly, we developed deep learning-based model for drug-target interactions prediction (Chapter 5). The constructed architecture exploits the topological structure of drug molecules, along with modeling spatio-sequential information. More specifically, we integrate three components that enable us to learn low-dimension vector representations for a target that is a protein sequence, and vector representations for a chemical compound.

6.2 Perspectives

In this Thesis we motivated, proposed and investigated a few novel solutions to the problems related to cheminformatics that focus on different aspects of representation learning. While our methodologies bring progress, important open challenges remain.

First of all, we think that future work on explanatory techniques would be necessary for proper usage of the presented approaches in practice. More investigation into how explanations can provide a complete view of the output returned by the models would also be valuable. In addition, robustifying setting-prediction to provide faithful insight into model's output generation process would also be an interesting direction of research.

Secondly, the introduced algorithms can further be improved. A possible future direction could be the use of the few-shot learning concept that was widely explored in computer vision [50, 92]. Specifically, the idea would be to pre-train a graph neural network to learn molecular embeddings. Then, a meta-learning framework could be developed to transfer knowledge from various prediction tasks and get a well-initialized model which could be fast adapted to work with few-shot data samples. Moreover, it would be worthwhile exploring architectures that combine both graph and sequence model to learn representation for the meta-learning process.

Another challenge posed for deep learning-based research in drug discovery research is the lack of data for reliable model development. Due to the fact that the wet-lab experiments in this area are time-consuming and expensive, most of the research only focuses on limited available data sets. One way to tackle this problem is to use reliable and widely validated simulation tools to generate huge amount of data.

All of these can be seen as challenges with the potential to not only keep scientists engaged for the years and decades to come, but also to show that the strength and competence in science combined with academic breadth and interdisciplinarity may benefit society as a whole.

Appendix A

Background: Deep Learning

> On being asked, “How is Perceptron performing today?” I am often tempted to respond, “Very well, thank you, and how are Neutron and Electron behaving?”
>
> —*Frank Rosenblatt*

This chapter first presents a thorough introduction to the most relevant classes of deep learning models to build a ground for our work. In this context, we start with discussing state-of-the-art feed forward architectures and temporal neural models. Then, the milestones of supervised learning are briefly discussed.

A.1 From shallow neural networks to deep architectures

Neural Networks (NNs) have dramatically improved the state-of-the-art in various different artificial intelligence tasks. Moreover, they have changed our daily life. Take for instance conversational interfaces/assistants such as Apple’s Siri, Amazon Alexa, Microsoft’s Cortana or Google Assistant whose presence helps users in many activities, including making phone calls, playing music and shopping online. In practice, the impact of neural networks is noticeable everywhere. They were applied to several application problems such as text-to-image [123] or text-to-speech [56, 139] processing. Greenspan et al. [63] presents remarkable improvements thanks to neural networks in medical imaging. For instance, Ben-Cohen et al. used this methodology for liver segmentation and lesions detection [10]. Neural Networks have become relevant for the more general fields of image processing [135, 171], computer vision (CV) [27], natural language processing (NLP) [43, 173], autonomous driving [28, 105], face recognition [134, 42], text understanding [84], art [83], optics [64], speech recognition [62], acoustic modeling [111], or anomaly detection [96]. To summarize, there is a strong push toward developing better architectures based on neural networks that have demonstrated vital success in the science and industry.

The history of neural networks began in the mid 1960's, and many people have contributed toward their development over the years. In fact, the advent of this technique is inspired by the architecture and the dynamics of networks of neurons in the mammalian brain [107]. For instance, the history of convolutional neural networks that are discussed in Section A.4 dates back to the mid-twentieth century when two major cell types in the primary visual cortex of cats were discovered by Hubel and Wiesel [76, 77]. They observed that complex cells receive input from many simple cells and thus have more spatially invariant responses. However, while some scientists were focusing on the biological system in the natural environment, in 1958 Rosenblatt introduced the Perceptron with the context of the vision system [127]. However, working with perceptron had been found to be difficult in practice by the late 1980. As Minsky and Papert highlighted in their work [110], perceptron is limited to represent only linear decision boundaries, but not XOR. Therefore, traditional machine learning algorithms, such as kernel machines [153, 133], have become increasingly successful and popular. Indeed, Rumelhart et al. demonstrated in their paper that perceptrons can be trained by gradient descent as late as in 1986. This was considered as a breakthrough in perceptron learning. Then, there were some early successes of neural networks with more layers, but these methods were not widely used. In the new millennium, deep neural networks (DNNs) have finally attracted wide-spread attention, mainly by the work of Hinton et al. [72]. The authors empirically proved that the proper initialization of DNNs enables to find good solutions in a reasonable amount of time. This was soon followed by a few deep neural network-based algorithms that have won many official international competitions. The next major demonstration of the power of DNNs came in 2012 when it was shown that a neural network trained using over a million real-world images [41] could perform classification into one of a thousand object categories [95].

A.2 Neural Networks as Function Approximators

In our work models that can be formulated as differentiable functions $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$ are taken into consideration. Specifically, the goal is to find this kind of functions by learning parameters θ from a set of training examples $\mathcal{T} = \{(x_i, y_i)\}$, where $x_i \in \mathcal{X}$ denotes the input and $y_i \in \mathcal{Y}$ some desired output of the ith training sample. For example, x_i could be a molecule represented by its chemical formula, and y_i may be a corresponding bioactivity score (e.g. true or false).

Also, a loss function is defined $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \times \Theta \to \mathbb{R}$ to measure the gap between a provided output y and a predicted output $f_\theta(x)$ on an observation x, given a current setting of parameters θ. To be more precise, we want to find those parameters θ^* that minimize the discrepancy on a training set. This leads to the following mathematical formula of our learning problem:

$$\theta^* = \arg\min_{\theta} \frac{1}{\mathcal{T}} \sum_{(x,y)\in\mathcal{T}} \mathcal{L}(f_\theta(x), y, \theta). \tag{A.1}$$

Here, we might not only want to measure the gap between given and predicted outputs, but also use a so-called regularizer on the parameters to improve generalization. Therefore, $\mathcal{L}$ is also a function of θ. Please note that in the remainder of this Thesis, we often omit θ in $\mathcal{L}$ for clarity. In addition, since both $\mathcal{L}$ and f_θ are differentiable functions, it is possible to use gradient-based optimization methods including Stochastic Gradient Descent (SGD) [124] and its numerous variants [122, 93] to iteratively update θ based on mini-batches $B \subseteq \mathcal{T}$ of the training samples. This step can be defined as:

$$\theta_{t+1} = \theta_t - \eta \Delta_{\theta_t} \frac{1}{\mathcal{B}} \mathcal{L}(f_\theta(x), y, \theta_t), \tag{A.2}$$

where Δ_θ is the differentiation operation of the loss with respect to parameters for the current batch at time step t, and η refers to a learning rate.

In addition, in order to be able to learn from data, one has to calculate the gradient of a loss with respect to all parameters that appear in the given model. Since it is assumed that all operations in the computation graph are differentiable, the Chain Rule of calculus can be recursively applied.

A Chain Rule in the calculus Let $z = g(y) = g(f(x))$ with $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $f : \mathbb{R}^l \rightarrow \mathbb{R}^n$ be a composite function. Then, the chain rule is employed to decompose the calculation of $\Delta_x z$, i.e., the gradient of the entire computation z with respect to x. Thus, the calculation takes the form:

$$\Delta_x z = (\frac{\partial y}{\partial x})^T \Delta_y z. \tag{A.3}$$

Please note that $\frac{\partial y}{\partial x}$ refers to the Jacobian matrix of f, which is the matrix of partial derivatives. $\Delta_y z$ denotes the gradient of z with respect to y.

A.3 Shallow neural networks

In general, a shallow neural network consists of a linear operation followed by a nonlinear function. More precisely, the basic idea of a neuron model is that we have an input data $x \in \mathbb{R}^{n_0}$, a bias $b \in \mathbb{R}^{n_1}$, the weights $W \in \mathbb{R}^{n_1 \times n_0}$, and a non-linear function $g(\cdot)$. During the training procedure the parameters are updated with the goal of minimizing a pre-defined loss function $\mathcal{L}(\hat{y}, y)$, such as the cross entropy loss. Figure A.1A provides a graphical interpretation of the computational unit employed in neural networks.

However, in order to build neural networks, the units need to be connected with each other in a layered structure that allows the input data to be gradually processed as it propagates through the network. Thus, the simplest architecture is a feedforward structure. In addition, to update the parameters a backpropagation process is used, as shown in Figure A.1B. In turn, backpropagation employs the Chain Rule A.2 to recursively define the efficient calculation of gradients of parameters and inputs in the computation graph. This allows to avoid recalculation of previously calculated expressions.

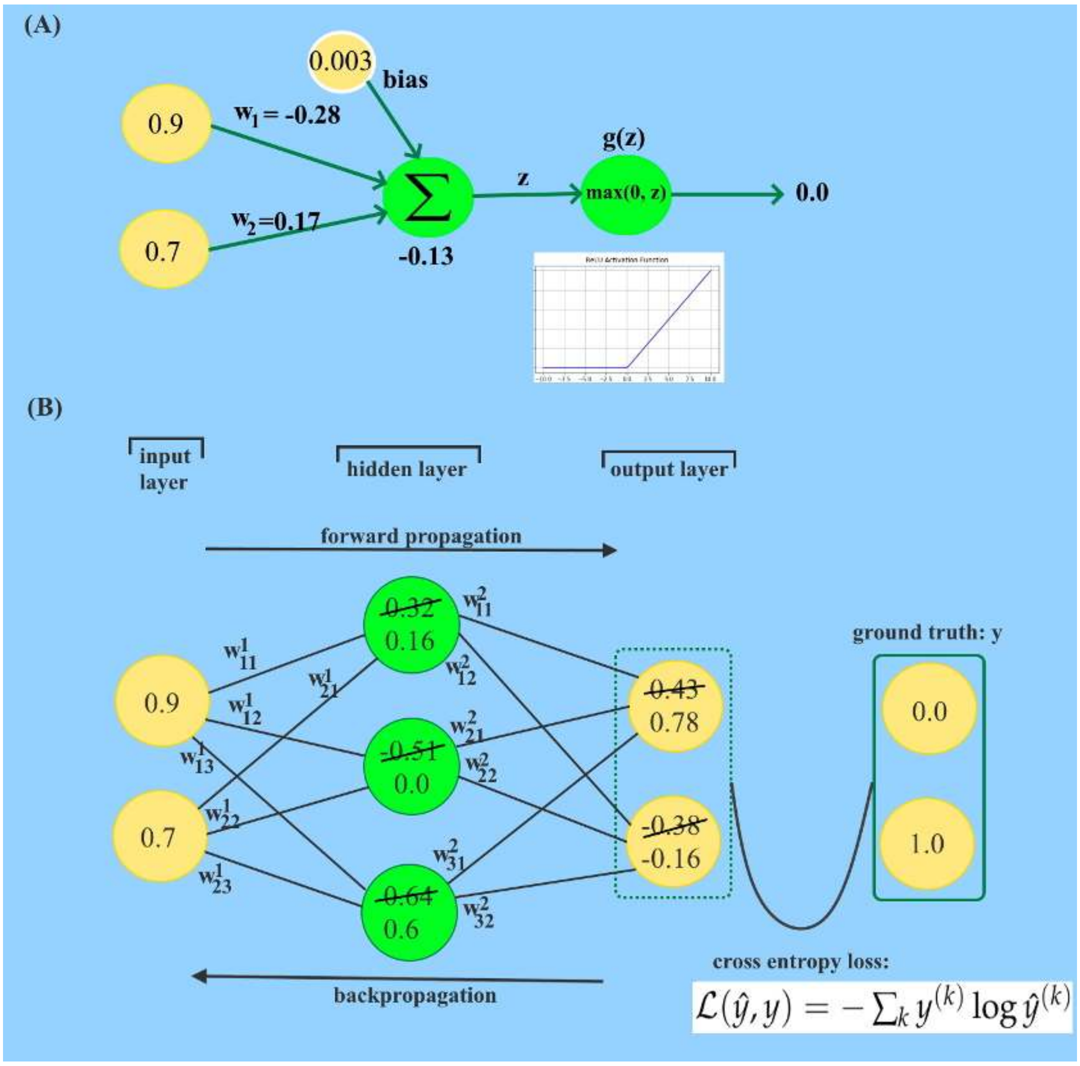


FIGURE A.1: Graphical visualization of a shallow neural network.

Additionally, depth and width of a neural network are the structure properties that need to be mentioned since their understanding is one of the central problems in the study of deep learning theory. Notably, the concept of depth of a network refers to the number of nonlinear transformations between the separating layers. In turn, by the dimensionality of a layer k, we mean the number of neurons. Furthermore, as Bengio indicated, the architectures with more hidden layers are called deep [11]. In addition, the non-linear functions (known as *activation functions*) that are applied for each layer introduce the non-linear property. The most popular functions are Sigmoid, Hyperbolic

TABLE A.1: An overview of frequently used activation functions.

Activation function	$g(z)$
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$
Hyperbolic tangent	$\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
ReLu	$R(z) = \max(0, z)$
Signum	$sgn(z) = \begin{cases} 1, & \text{for } z > 1 \\ 0, & \text{for } z = 0 \\ -1, & \text{for } z < 0 \end{cases}$
Softmax	$f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$
ELU	$ELU(z) = \begin{cases} \alpha(e^{z} - 1), & \text{for } z \leq 0 \\ z, & \text{for } z > 0 \end{cases}$

tangent, and Rectified linear unit (ReLU). An overview of frequently used activation functions is given in Table A.1.

Here, we have to explain that, at first, the reasons behind using the shallow neural networks seem to be quite rational. Nevertheless, there are several disadvantages associated with shallow neural networks. One of the most vital is an enormous number of parameters. Let us imagine that a one hidden layer has k_1 nodes and the subsequent output layer contains k_2 nodes. As a result, we would have $k_1 * k_2$ parameters between those two layers. This usually leads to *overfitting* [68] and the model is unable to generalize well on the testing data. Also, the slowing down of the training and testing process is becoming a serious problem. Another drawback of shallow neural networks is the fact that they take into consideration each input feature independently. In consequence, the correlation between input features tend to be ignored. However, this is an important concern in the context of chemical data.

Please also note that although a feedforward network is capable of approximating any smooth function, one does not have guarantees that this approximation can be really learned. This is caused by a few factors but the most important are overfitting and specific properties of currently used optimization algorithms. Also, as mentioned above, there is no theorem that indicates the particular number of hidden units required for such an approximation. Thus, in the worst-case setting, if we want to memorize every possible input, this number may be even exponential. This causes that deeper architectures are usually chosen. Furthermore, it shows that a great families of functions may be approximated in a much more compact way if the depth of the network is greater than a given value of the depth.

One another important fact about deep neural network design is that the representation that interests us (also called network embedding) is the intermediate output at the end of the specific model. Equation A.4 defines how it may be obtained:

$$f(x, W^{(1)}, \dots, W^{(k)}, b^{(1)}, \dots, b^{(k)}) = h(W^{(k)} \dots h(W^{(2)} h(W^{(1)} x + b^{(1)}) + b^{(2)}) \dots + b^{(k)}), \tag{A.4}$$

where $h(\cdot)$ and $f(\cdot)$ are activation functions. In turn, the final output of the model, such as a prediction value, is calculated from the obtained representation using an extra task driven layer or a stack of layers.

A.4 Convolutional neural networks

Convolutional neural network (CNN) can be treated as a type of feedforward neural network that uses convolution structures in at least one of its layers to extract features from data. Therefore, unlike the traditional feature extraction algorithms, CNN does not extract features with the help of feature engineering. This class of models turns out to be a suitable tool for processing signals in tensor forms where the tensor elements are arranged in a meaningful order. In other words, CNNs is supposed to work well if patterns in the data are treated under the assumption of temporal or spatial invariance. The examples include speech, images, or video but are not limited to.

Specifically, to construct a convolutional neural network-based architecture, one needs at least four building blocks. The procedure of a CNN is shown in Figure A.2. As the name suggests, the core operation is convolution that aims at feature extraction. As a result, we obtain an output feature map. A convolution matrix convolves with the object such as an image using a given set of weights and multiplying its elements with the corresponding elements of the small slices. The idea underlying the convolutional layer operation is shown in Figure A.3. Since the kernel has a certain size, the information in the border may be lost. For this reason a padding operation is employed that helps to enlarge the input with zeros, and indirectly adjust the size. We also have to adjust the stride to control the density of convolution. Obviously, the smaller the value of stride, the higher the density. As one may notice, the final feature map can be comprised of too many features leading to overfitting problem. To deal with this issue, pooling [53], namely down-sampling, in the form of average pooling and max pooling is introduced. In the end, a one or more fully-connected layers appear to perform the same duties found in shallow neural networks and to produce scores from the activations to be used for classification or regression.

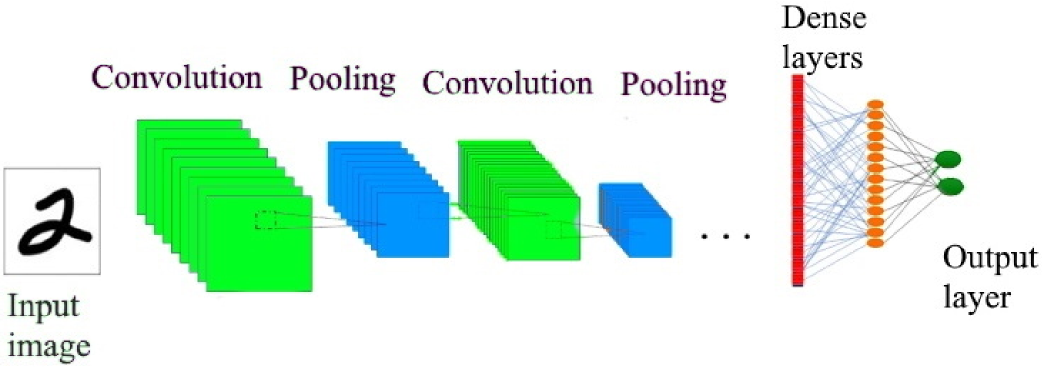
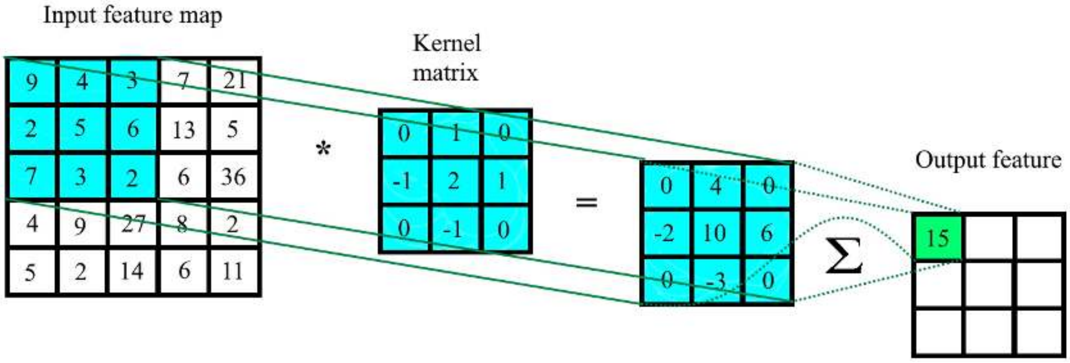


FIGURE A.2: An example of a CNN architecture with 2 convolution stages.



$$\begin{bmatrix} 9 & 4 & 3 & 7 & 21 \\ 2 & 5 & 6 & 13 & 5 \\ 7 & 3 & 2 & 6 & 36 \\ 4 & 9 & 27 & 8 & 2 \\ 5 & 2 & 14 & 6 & 11 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ -1 & 2 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 4 & 0 \\ -2 & 10 & 6 \\ 0 & -3 & 0 \end{bmatrix} \quad \Sigma \quad \begin{bmatrix} 15 & & \\ & & \\ & & \end{bmatrix}$$

FIGURE A.3: A visual representation of a convolutional layer.

A.5 Graph neural networks

In recent years, inspired by the success of CNNs, new theorems and definitions of vital operations have been developed to deal with the complexity of graph data. For instance, one can define a general concept of graph convolutions based on a 2D convolution (see Figure A.4). This Section outlines the background of graph neural networks (GNNs).

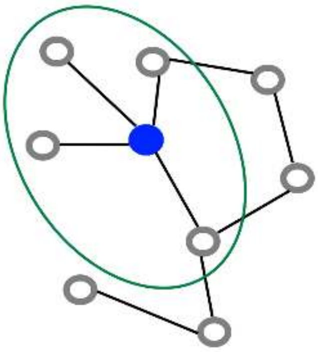


FIGURE A.4: Graph convolution. In order to obtain a representation of the blue vertex, a solution based on the graph convolutional operation would be to take the average value of the vertex features of the blue vertex along with its direct neighbors.

The concept of GNN model was first given by Gori et al. in 2005 [61], and later generalized by Scarselli et al. [131] in 2008 and Gallicchio et al. [54] in 2010. The idea is as follows. At the beginning, every vertex has an initial representation given by its features $\{h_i \in \mathbb{R}^d | i \in \mathcal{V}\}$ and the set of edges $\mathcal{E}$. At each propagation step: nodes create a message with their embedding, send the message to the neighbors, receive, collect and aggregate messages, and update their embedding using the received information. This operation is repeated until convergence. The final embeddings are employed for prediction. More formally, a layer returns a new set of vertex representations $\{h' \in \mathbb{R}^{d'} | i \in \mathcal{V}\}$, where the same parameter guided function is applied to every vertex given its direct neighbors $N(i) = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$.

$$h'_i = f_\theta(h_i, AGGREGATE(\{h_j | j \in N(i)\})). \tag{A.5}$$

In general, there are many variants of paradigms of GNNs. The examples are Graph Recurrent Neural Networks [143], Graph Convolutional Networks [19, 48, 94, 5], Graph Generative Networks [151], Spatial-Temporal Graph Neural Networks [100, 174, 136, 168], and various different hybrid forms [38, 44]. The next two parts of this Section is discuss the issue of Graph Convolutional Networks and Graph Attention Networks since they will be one of the most important subjects of the main part of this Thesis. Indeed, the design of f and $AGGREGATE$ is what mainly distinguishes one type of GNN from the other.

A.5.1 Graph Convolutional Networks

GCN-like models are comprised of so-called aggregators and updaters. The role of aggregator is to gather information based on the graph structure, while the updater aims at updating nodes' hidden states according to the obtained information. To be more precise, the following equation defines this operation:

$$H^{(l+1)} = \sigma(LH^{(l)}W^{(l)}), \tag{A.6}$$

where d denotes the number of features, $L \in \mathbb{R}^{n \times n}$ refers to the aggregation matrix, $H^{(l)} = (h_1^{(l)}, \dots, h_N^{(l)})^T \in \mathbb{R}^{n \times d^{(l)}}$ denotes the node representation matrix in lth layer, $H^{(0)} = X$, $W^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l+1)}}$ is the trainable weight matrix in lth layer, and $\sigma(\cdot)$ refers to the activation function such as ReLU, LeakyReLU, or ELU. Moreover, GCN's aggregator is based on the re-normalized graph Laplacian $\hat{A}$:

$$L = \hat{A} \approx \tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5}, \tag{A.7}$$

where $\tilde{A} = A + I_n$, and $\tilde{D}$ denotes the diagonal matrix of node degrees with $D_{ii} = \sum_{j=1}^{n} \hat{a_{ij}}$.

A.5.2 Graph Attention Networks

According to the GCNs approach, the neighborhours of vertex i, i.e., $j \in N(i)$ are aggregated with equal or predefined weights. In fact, as one may notice, the impact of neighbors may not be the same and depend on the given neighbour [79]. Therefore, to tackle this issue, in GAT the Equation A.5 is modified, and a learned weighted average of the representations of $N(i)$ is computed. Specifically, we employ a scoring function $e : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to compute a score for every edge (j, i). Such an approach allows to indicate how important are the features of the neighbor j to the node i:

$$e(h_i, h_j) = LeakyReLU(a^T \cdot [Wh_i || Wh_j]), \tag{A.8}$$

where $\alpha \in \mathbb{R}^{2d'}$, $W \in \mathbb{R}^{d' \times d}$ are learned, and $||$ is a concatenation operator. These scores, namely attention scores, are normalized across all neighbors $j \in N(i)$ with softmax function. In consequence, the attention function is given by:

$$\alpha_{ij} = softmax_j(e(h_i, h_j)) = \frac{\exp(e(h_i, h_j))}{\sum_{j' \in N(i)} \exp(e(h_i, h_{j'}))}. \tag{A.9}$$

Next, the new representation of i is obtained by computing a weighted average of the transformed features of the neighbor vertices (followed by a non-linearity σ) using the normalized attention coefficients as follows:

$$h_i' = \sigma(\sum_{j \in N(i)} \alpha_{ij} \cdot Wh_j). \tag{A.10}$$

As a result, Equations A.8 and A.10 form the definition of GAT.

A.6 Remarks on supervised learning

The goal of supervised learning is to learn a predictor for a task having ratings, values, labels, etc. In our case, it we want to learn a function, i.e., a classifier or a predictor f. Therefore, in the training step, the the model's objective is, given $(x_i)_{i\in[1,\ldots,n]}$ and $(y_i)_{i\in[1,\ldots,n]}$, make an attempt to match $(x_i, f(x_i))$ with (x_i, y_i) as accurately as possible. Then, the learned function is used to map new samples. However, the core challenge here is to propose a function that generalizes well from labeled to unlabeled data.

Bibliography

[1] Mohamed Abd Elaziz et al. “Toxicity risks evaluation of unknown FDA biotransformed drugs based on a multi-objective feature selection approach”. In: *Applied Soft Computing* (2019), p. 105509.

[2] Ahmed Abdelaziz et al. “Consensus modeling for HTS assays using in silico descriptors calculates the best balanced accuracy in Tox21 challenge”. In: *Frontiers in Environmental Science* 4 (2016), p. 2.

[3] National Center for Advancing Translational Sciences. *Tox21 Data Challenge 2014*. 2014. URL: `https://tripod.nih.gov/tox21/challenge/`.

[4] Han Altae-Tran et al. “Low data drug discovery with one-shot learning”. In: *ACS central science* 3.4 (2017), pp. 283–293.

[5] James Atwood and Don Towsley. “Diffusion-convolutional neural networks”. In: *Advances in neural information processing systems*. 2016, pp. 1993–2001.

[6] Amos Bairoch and Rolf Apweiler. “The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000”. In: *Nucleic acids research* 28.1 (2000), pp. 45–48.

[7] II Baskin et al. “Multilevel approach to the prediction of properties of organic compounds in the framework of the QSAR/QSPR methodology”. In: *Doklady Chemistry*. Vol. 427. 1. SP MAIK Nauka/Interperiodica. 2009, pp. 172–175.

[8] Richard Bellman. "Dynamic programming". In: *Science* 153.3731 (1966), pp. 34–37.

[9] Guy W Bemis and Mark A Murcko. "The properties of known drugs. 1. Molecular frameworks". In: *Journal of medicinal chemistry* 39.15 (1996), pp. 2887–2893.

[10] Avi Ben-Cohen et al. "Fully convolutional network for liver segmentation and lesions detection". In: *Deep learning and data labeling for medical applications*. Springer, 2016, pp. 77–85.

[11] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[12] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[13] Esben Jannik Bjerrum and Boris Sattarov. "Improving chemical autoencoder latent space and molecular de novo generation diversity with heteroencoders". In: *Biomolecules* 8.4 (2018), p. 131.

[14] Burton H Bloom. "Space/time trade-offs in hash coding with allowable errors". In: *Communications of the ACM* 13.7 (1970), pp. 422–426.

[15] Regine S Bohacek, Colin McMartin, and Wayne C Guida. "The art and practice of structure-based drug design: a molecular modeling perspective". In: *Medicinal research reviews* 16.1 (1996), pp. 3–50.

[16] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. *Feature selection for high-dimensional data*. Springer, 2015.

[17] Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.

[18] Frank K Brown et al. "Chemoinformatics: what is it and how does it impact drug discovery". In: *Annual reports in medicinal chemistry* 33 (1998), pp. 375–384.

[19] Joan Bruna et al. “Spectral networks and locally connected networks on graphs”. In: *arXiv preprint arXiv:1312.6203* (2013).

[20] Peter Buchwald and Nicholas Bodor. “Computer-aided drug design: the role of quantitative structure–property, structure–activity and structure–metabolism relationships (QSPR, QSAR, QSMR)”. In: *Drugs Future* 27.6 (2002), pp. 577–588.

[21] Frank R Burden and David A Winkler. “New QSAR methods applied to structure- activity mapping and combinatorial chemistry”. In: *Journal of chemical information and computer sciences* 39.2 (1999), pp. 236–242.

[22] Gaspar Cano et al. “Automatic selection of molecular descriptors using random forest: Application to drug discovery”. In: *Expert Systems with Applications* 72 (2017), pp. 151–159.

[23] D-S Cao et al. “In silico toxicity prediction by support vector machine and SMILES representation-based string kernel”. In: *SAR and QSAR in Environmental Research* 23.1-2 (2012), pp. 141–153.

[24] Dong-Sheng Cao et al. “ChemoPy: freely available python package for computational biology and chemoinformatics”. In: *Bioinformatics* 29.8 (2013), pp. 1092–1094.

[25] G Cerruela Garcıa et al. “Molecular activity prediction by means of supervised subspace projection based ensembles of classifiers”. In: *SAR and QSAR in Environmental Research* 29.3 (2018), pp. 187–212.

[26] Swapnil Chavan, Ran Friedman, and Ian A Nicholls. “Acute toxicity-supported chronic toxicity prediction: a k-nearest neighbor coupled read-across strategy”. In: *International journal of molecular sciences* 16.5 (2015), pp. 11659–11677.

[27] Hongming Chen et al. “The rise of deep learning in drug discovery”. In: *Drug discovery today* 23.6 (2018), pp. 1241–1250.

[28] Zhe Chen, Jing Zhang, and Dacheng Tao. "Progressive lidar adaptation for road detection". In: *IEEE/CAA Journal of Automatica Sinica* 6.3 (2019), pp. 693–702.

[29] Artem Cherkasov et al. "QSAR modeling: where have you been? Where are you going to?" In: *Journal of medicinal chemistry* 57.12 (2014), pp. 4977–5010.

[30] Stefan Chmiela et al. "Machine learning of accurate energy-conserving molecular force fields". In: *Science advances* 3.5 (2017), e1603015.

[31] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[32] Kangway V Chuang, Laura Gunsalus, and Michael J Keiser. "Learning Molecular Representations for Medicinal Chemistry". In: *Journal of Medicinal Chemistry* (2020).

[33] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine learning* 20.3 (1995), pp. 273–297.

[34] George E Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. "Multi-task neural networks for QSAR predictions". In: *arXiv preprint arXiv:1406.1231* (2014).

[35] Sujata Dash et al. *Deep Learning Techniques for Biomedical and Health Informatics*. Springer, 2020.

[36] Laurianne David et al. "Applications of deep-learning in exploiting large-scale and heterogeneous compound data in industrial pharmaceutical research". In: *Frontiers in pharmacology* 10 (2019).

[37] Mindy I Davis et al. "Comprehensive analysis of kinase inhibitor selectivity". In: *Nature biotechnology* 29.11 (2011), pp. 1046–1051.

[38] Nicola De Cao and Thomas Kipf. "MolGAN: An implicit generative model for small molecular graphs". In: *arXiv preprint arXiv:1805.11973* (2018).

[39] "Dealing with a data-limited regime: Combining transfer learning and transformer attention mechanism to increase aqueous solubility prediction performance". In: *Artificial Intelligence in the Life Sciences* 1 (2021), p. 100021. DOI: `https://doi.org/10.1016/j.ailsci.2021.100021`.

[40] John C Dearden. "The history and development of quantitative structure-activity relationships (QSARs)". In: *Oncology: breakthroughs in research and practice*. IGI Global, 2017, pp. 67–117.

[41] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[42] Jiankang Deng et al. "Arcface: Additive angular margin loss for deep face recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4690–4699.

[43] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[44] Kien Do, Truyen Tran, and Svetha Venkatesh. "Graph transformation policy network for chemical reaction prediction". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 750–760.

[45] Pedro Domingos. "A few useful things to know about machine learning". In: *Communications of the ACM* 55.10 (2012), pp. 78–87.

[46] Cyril Dominguez, Rolf Boelens, and Alexandre MJJ Bonvin. "HADDOCK: a protein- protein docking approach based on biochemical or biophysical information". In: *Journal of the American Chemical Society* 125.7 (2003), pp. 1731–1737.

[47] DL Donoho and M Elad. "Maximal sparsity representation via l1 minimization". In: *Proceedings of National Academy of Sciences* 100 (2003), pp. 2197–2202.

[48] David Duvenaud et al. “Convolutional networks on graphs for learning molecular fingerprints”. In: *arXiv preprint arXiv:1509.09292* (2015).

[49] Yanghe Feng, Qi Wang, and Tengjiao Wang. “Drug target protein-protein interaction networks: a systematic perspective”. In: *BioMed research international* 2017 (2017).

[50] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-agnostic meta-learning for fast adaptation of deep networks”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1126–1135.

[51] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[52] Toshio Fujita and David A Winkler. “Understanding the roles of the “two QSARs””. In: *Journal of chemical information and modeling* 56.2 (2016), pp. 269–274.

[53] Kunihiko Fukushima and Sei Miyake. “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”. In: *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[54] Claudio Gallicchio and Alessio Micheli. “Graph Echo State Networks”. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. 2010, pp. 1–8. DOI: 10.1109/IJCNN.2010.5596796.

[55] Kaifu Gao et al. “Are 2D fingerprints still valuable for drug discovery?” In: *Physical Chemistry Chemical Physics* 22.16 (2020), pp. 8373–8390.

[56] Andrew Gibiansky et al. “Deep Voice 2: Multi-Speaker Neural Text-to-Speech.” In: *NIPS*. 2017.

[57] Justin Gilmer et al. “Neural message passing for quantum chemistry”. In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272.

[58] Garrett B Goh et al. "Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed QSAR/QSPR models". In: *arXiv preprint arXiv:1706.06689* (2017).

[59] Garrett B. Goh et al. "SMILES2Vec: An Interpretable General-Purpose Deep Neural Network for Predicting Chemical Properties". In: *CoRR* abs/1712.02034 (2017). arXiv: `1712.02034`. URL: `http://arxiv.org/abs/1712.02034`.

[60] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[61] Marco Gori, Gabriele Monfardini, and Franco Scarselli. "A new model for learning in graph domains". In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 2. IEEE. 2005, pp. 729–734.

[62] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, pp. 6645–6649.

[63] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1153–1159.

[64] Harel Haim et al. "Depth estimation from a single image using deep learned phase coded mask". In: *IEEE Transactions on Computational Imaging* 4.3 (2018), pp. 298–310.

[65] William L Hamilton, Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 1025–1035.

[66] Katja Hansen et al. “Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space”. In: *The journal of physical chemistry letters* 6.12 (2015), pp. 2326–2331.

[67] G Jean Harry et al. “In vitro techniques for the assessment of neurotoxicity.” In: *Environmental health perspectives* 106.suppl 1 (1998), pp. 131–158.

[68] Douglas M Hawkins. “The problem of overfitting”. In: *Journal of chemical information and computer sciences* 44.1 (2004), pp. 1–12.

[69] Michael Hay et al. “Clinical development success rates for investigational drugs”. In: *Nature biotechnology* 32.1 (2014), pp. 40–51.

[70] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[71] Tong He et al. “SimBoost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines”. In: *Journal of cheminformatics* 9.1 (2017), pp. 1–14.

[72] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[73] Geoffrey E Hinton et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012).

[74] Andrew L Hopkins. “Predicting promiscuity”. In: *Nature* 462.7270 (2009), pp. 167–168.

[75] Weihua Hu et al. “Strategies for pre-training graph neural networks”. In: *arXiv preprint arXiv:1905.12265* (2019).

[76] David H Hubel and Torsten N Wiesel. "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of physiology* 148.3 (1959), pp. 574–591.

[77] David H Hubel and Torsten N Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of physiology* 160.1 (1962), pp. 106–154.

[78] James P Hughes et al. "Principles of early drug discovery". In: *British journal of pharmacology* 162.6 (2011), pp. 1239–1249.

[79] Vassilis N Ioannidis, Antonio G Marques, and Georgios B Giannakis. "Tensor graph convolutional networks for multi-relational and robust learning". In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 6535–6546.

[80] Katsuhiko Ishiguro, Kenta Oono, and Kohei Hayashi. "Weisfeiler-Lehman embedding for molecular graph neural networks". In: *arXiv preprint arXiv:2006.06909* (2020).

[81] Sabrina Jaeger, Simone Fulle, and Samo Turk. "Mol2vec: unsupervised machine learning approach with chemical intuition". In: *Journal of chemical information and modeling* 58.1 (2018), pp. 27–35.

[82] CA James, D Weininger, and J Delany. "Daylight theory manual. daylight chemical information systems". In: *Inc., Irvine, CA* (1995).

[83] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European conference on computer vision*. Springer. 2016, pp. 694–711.

[84] Rudolf Kadlec et al. "Text understanding with the attention sum reader network". In: *arXiv preprint arXiv:1603.01547* (2016).

[85] Yoshiki Kato, Shinji Hamada, and Hitoshi Goto. "Molecular activity prediction using deep learning software library". In: *2016 international conference on advanced informatics: concepts, theory and application (ICAICTA)*. IEEE. 2016, pp. 1–6.

[86] Kentaro Kawai, Satoshi Fujishima, and Yoshimasa Takahashi. "Predictive activity profiling of drugs by topological-fragment-spectra-based support vector machines". In: *Journal of chemical information and modeling* 48.6 (2008), pp. 1152–1160.

[87] Steven Kearnes, Brian Goldman, and Vijay Pande. "Modeling industrial ADMET data with multitask networks". In: *arXiv preprint arXiv:1606.08793* (2016).

[88] Steven Kearnes et al. "Molecular graph convolutions: moving beyond fingerprints". In: *Journal of computer-aided molecular design* 30.8 (2016), pp. 595–608.

[89] Michael J Keiser et al. "Relating protein pharmacology by ligand chemistry". In: *Nature biotechnology* 25.2 (2007), pp. 197–206.

[90] Asad U Khan et al. "Descriptors and their selection methods in QSAR analysis: paradigm for drug design". In: *Drug discovery today* 21.8 (2016), pp. 1291–1302.

[91] Lemont Kier. *Molecular orbital theory in drug research*. Vol. 10. Elsevier, 2012.

[92] Jongmin Kim et al. "Edge-labeling graph neural network for few-shot learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11–20.

[93] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[94] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[95] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[96] Donghwoon Kwon et al. "A survey of deep learning-based network anomaly detection". In: *Cluster Computing* 22.1 (2019), pp. 949–961.

[97] Greg Landrum. "RDKit: Open-Source Cheminformatics Software". In: (2016). URL: https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

[98] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[99] Eelke B Lenselink et al. "Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set". In: *Journal of cheminformatics* 9.1 (2017), pp. 1–14.

[100] Yaguang Li et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting". In: *arXiv preprint arXiv:1707.01926* (2017).

[101] Angelica Nakagawa Lima et al. "Use of machine learning approaches for novel drug discovery". In: *Expert opinion on drug discovery* 11.3 (2016), pp. 225–239.

[102] Shengchao Liu et al. "Practical model selection for prospective virtual screening". In: *Journal of chemical information and modeling* 59.1 (2018), pp. 282–293.

[103] Tiqing Liu et al. "BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities". In: *Nucleic acids research* 35.suppl_1 (2007), pp. D198–D201.

[104] Junshui Ma et al. "Deep neural nets as a method for quantitative structure–activity relationships". In: *Journal of chemical information and modeling* 55.2 (2015), pp. 263–274.

[105] Wei-Chiu Ma et al. "Deep rigid instance scene flow". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3614–3622.

[106] Magdalena Wiercioch oraz Marek Śmieja. "Mixture of Metrics Optimization for Machine Learning Problems". In: *Schedae Informaticae*. Vol. 2015. Volume 24. 2016.

[107] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[108] David Mendez et al. “ChEMBL: towards direct deposition of bioassay data”. In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D930–D940. ISSN: 0305-1048. DOI: 10.1093/nar/gky1075. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D930/27437436/gky1075.pdf. URL: https://doi.org/10.1093/nar/gky1075.

[109] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 3111–3119.

[110] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[111] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. “Acoustic modeling using deep belief networks”. In: *IEEE transactions on audio, speech, and language processing* 20.1 (2011), pp. 14–22.

[112] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[113] Chanin Nantasenamat, Chartchalerm Isarankura-Na-Ayudhya, and Virapong Prachayasittikul. “Advances in computational methods to predict the biological activity of compounds”. In: *Expert opinion on drug discovery* 5.7 (2010), pp. 633–654.

[114] † Nidhi et al. “Prediction of biological targets for compounds using multiple-category Bayesian models trained on chemogenomics databases”. In: *Journal of chemical information and modeling* 46.3 (2006), pp. 1124–1133.

[115] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. "DeepDTA: deep drug–target binding affinity prediction". In: *Bioinformatics* 34.17 (2018), pp. i821–i829.

[116] Hakime Öztürk, Elif Ozkirimli, and Arzucan Özgür. "A novel methodology on distributed representations of proteins using their interacting ligands". In: *Bioinformatics* 34.13 (2018), pp. i295–i303.

[117] Tapio Pahikkala et al. "Toward more realistic drug–target interaction predictions". In: *Briefings in bioinformatics* 16.2 (2015), pp. 325–337.

[118] Arindam Paul et al. "Chemixnet: Mixed dnn architectures for predicting chemical properties using multiple molecular representations". In: *arXiv preprint arXiv:1811.08283* (2018).

[119] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[120] Pavel G Polishchuk et al. "Application of random forest approach to QSAR prediction of aquatic toxicity". In: *Journal of chemical information and modeling* 49.11 (2009), pp. 2481–2488.

[121] Kristina Preuer et al. "Interpretable deep learning in drug discovery". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 331–345.

[122] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. "Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization". In: ICML'12. Omnipress, 2012, pp. 1571–1578.

[123] Scott Reed et al. "Generative adversarial text to image synthesis". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1060–1069.

[124] Herbert Robbins and Sutton Monro. "A stochastic approximation method". In: *The annals of mathematical statistics* (1951), pp. 400–407.

[125] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5 (2010), pp. 742–754.

[126] David J Rogers and Taffee T Tanimoto. "A computer program for classifying plants". In: *Science* 132.3434 (1960), pp. 1115–1118.

[127] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[128] Kunal Roy. "Advances in QSAR modeling". In: *Applications in pharmaceutical, chemical, food, agricultural and environmental sciences. Springer, Cham* 555 (2017).

[129] Matthias Rupp et al. "Fast and accurate modeling of molecular atomization energies with machine learning". In: *Physical review letters* 108.5 (2012), p. 058301.

[130] Daniel Russo and James Zou. "How much does your data exploration overfit? controlling bias via information usage". In: *IEEE Transactions on Information Theory* 66.1 (2019), pp. 302–323.

[131] Franco Scarselli et al. "The graph neural network model". In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80.

[132] Petra Schneider et al. "Rethinking drug design in the artificial intelligence era". In: *Nature Reviews Drug Discovery* 19.5 (2020), pp. 353–364.

[133] Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, eds. *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999. ISBN: 0262194163.

[134] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[135] Eli Schwartz, Raja Giryes, and Alex M Bronstein. "DeepISP: Toward learning an end-to-end image processing pipeline". In: *IEEE Transactions on Image Processing* 28.2 (2018), pp. 912–923.

[136] Youngjoo Seo et al. “Structured sequence modeling with graph convolutional recurrent networks”. In: *International Conference on Neural Information Processing*. Springer. 2018, pp. 362–373.

[137] Arnab Seth and Kunal Roy. “QSAR modeling of algal low level toxicity values of different phenol and aniline derivatives using 2D descriptors”. In: *Aquatic Toxicology* 228 (2020), p. 105627.

[138] Marek Śmieja and Magdalena Wiercioch. “Constrained clustering with a complex cluster structure”. In: *Advances in Data Analysis and Classification* 11.3 (2017), pp. 493–518.

[139] Jose Sotelo et al. “Char2wav: End-to-end speech synthesis”. In: (2017).

[140] Iurii Sushko et al. “Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information”. In: *Journal of computer-aided molecular design* 25.6 (2011), pp. 533–554.

[141] Vladimir Svetnik et al. “Random forest: a classification and regression tool for compound classification and QSAR modeling”. In: *Journal of chemical information and computer sciences* 43.6 (2003), pp. 1947–1958.

[142] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[143] Kai Sheng Tai, Richard Socher, and Christopher D Manning. “Improved semantic representations from tree-structured long short-term memory networks”. In: *arXiv preprint arXiv:1503.00075* (2015).

[144] Jing Tang et al. “Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis”. In: *Journal of Chemical Information and Modeling* 54.3 (2014), pp. 735–743.

[145] Weihao Tang et al. “Deep learning for predicting toxicity of chemicals: A mini review”. In: *Journal of Environmental Science and Health, Part C* 36.4 (2018), pp. 252–271.

[146] Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*. Vol. 11. John Wiley & Sons, 2008.

[147] Roberto Todeschini and Viviana Consonni. *Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references*. Vol. 41. John Wiley & Sons, 2009.

[148] Katya Tsaioun and Steven A Kates. *ADMET for medicinal chemists: a practical guide*. John Wiley & Sons, 2011.

[149] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences". In: *Bioinformatics* 35.2 (2019), pp. 309–318.

[150] Muhammad Fahim Uddin et al. "Proposing enhanced feature engineering and a selection model for machine learning processes". In: *Applied Sciences* 8.4 (2018), p. 646.

[151] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. "Learning Localized Representations of Point Clouds With Graph-Convolutional Generative Adversarial Networks". In: *IEEE Transactions on Multimedia* 23 (2020), pp. 402–414.

[152] Han Van De Waterbeemd and Eric Gifford. "ADMET in silico modelling: towards prediction paradise?" In: *Nature reviews Drug discovery* 2.3 (2003), pp. 192–204.

[153] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[154] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[155] Yanli Wang et al. "Pubchem bioassay: 2017 update". In: *Nucleic acids research* 45.D1 (2017), pp. D955–D963.

[156] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[157] Magdalena Wiercioch. “Exploring the potential of spherical harmonics and PCVM for compounds activity prediction”. In: *International Journal of Molecular Sciences* 20.9 (2019), p. 2175.

[158] Magdalena Wiercioch. “Feature Selection in Texts”. In: *International Conference on Computer Recognition Systems*. Springer. 2017, pp. 336–345.

[159] Magdalena Wiercioch. “On Modeling Objects Using Sequence of Moment Invariants”. In: *Computer Information Systems and Industrial Management*. Ed. by Khalid Saeed and Władysław Homenda. Cham: Springer International Publishing, 2018, pp. 92–102. ISBN: 978-3-319-99954-8.

[160] Magdalena Wiercioch. “Towards learning word representation”. In: *Schedae Informaticae*. Vol. 25. 2016.

[161] Magdalena Wiercioch and Johannes Kirchmair. “Deep Neural Network Approach to Predict Properties of Drugs and Drug-Like Molecules”. In: *ML for Molecules Workshop at NeurIPS 2020*. 2020.

[162] Magdalena Wiercioch, Marek Śmieja, and Jacek Tabor. “Probability Index of Metric Correspondence as a measure of visualization reliability”. In: *ECML PKDD workshop on Machine Learning and Life Science (MLLS 2016)*. 2015.

[163] Peter Willett. “The calculation of molecular structural similarity: principles and practice”. In: *Molecular informatics* 33.6-7 (2014), pp. 403–413.

[164] David S Wishart et al. “DrugBank 5.0: a major update to the DrugBank database for 2018”. In: *Nucleic acids research* 46.D1 (2018), pp. D1074–D1082.

[165] William J Wiswesser. “107 Years of Line-Formula Notations (1861-1968)”. In: *Journal of Chemical Documentation* 8.3 (1968), pp. 146–150.

[166] Kedi Wu and Guo-Wei Wei. "Quantitative toxicity prediction using topology based multitask deep neural networks". In: *Journal of chemical information and modeling* 58.2 (2018), pp. 520–531.

[167] Zhenqin Wu et al. "MoleculeNet: a benchmark for molecular machine learning". In: *Chemical science* 9.2 (2018), pp. 513–530.

[168] Zonghan Wu et al. "Graph WaveNet for Deep Spatial-Temporal Graph Modeling". In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. Ed. by Sarit Kraus. ijcai.org, 2019, pp. 1907–1913. DOI: 10.24963/ijcai.2019/264. URL: https://doi.org/10.24963/ijcai.2019/264.

[169] Keyulu Xu et al. "How powerful are graph neural networks?" In: *arXiv preprint arXiv:1810.00826* (2018).

[170] Zheng Xu et al. "Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery". In: *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*. 2017, pp. 285–294.

[171] Wenming Yang et al. "Deep learning for single image super-resolution: A brief review". In: *IEEE Transactions on Multimedia* 21.12 (2019), pp. 3106–3121.

[172] Xin Yang et al. "Concepts of artificial intelligence for computer-assisted drug discovery". In: *Chemical reviews* 119.18 (2019), pp. 10520–10594.

[173] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *arXiv preprint arXiv:1906.08237* (2019).

[174] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting". In: *arXiv preprint arXiv:1709.04875* (2017).

[175] Ruisheng Zhang et al. "Neural network-molecular descriptors approach to the prediction of properties of alkenes". In: *Computers & chemistry* 21.5 (1997), pp. 335–341.

[176] Marinka Zitnik et al. “Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities”. In: *Information Fusion* 50 (2019), pp. 71–91.