Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Transfer learning in multitask drug-target interaction prediction using large-scale public datasets

Scientific Students' Association Report

Author:

Dániel Sándor

Advisor:

Dr. Péter Antal

2020

Contents

Kivonat

A gyógyszerkutatás területén egy új hatóanyag kifejlesztése vagy egy régi újrapozícionálása komoly erőforrásokat igényel mind pénzügyi, mind a ráfordított idő szempontjából. Az in silico gyógyszerkutatás ennek segítőjeként jelent meg, és mára egyre nagyobb jelentőséggel bír. A számítógépes modellek és erőforrások mellett ennek egyre fontosabb tényezője a nagy mennyiségű nyilvános adaton és tudás, valamint a gyógyszergyáraknál felgyűlt, korábbi kísérletek során létrejött, adat. Ezen adatok használatakor két fő nehézséggel kell számolni: a heterogenitással és a ritkasággal, azaz a lehetséges gyógyszerjelöltek terének hatalmas volta miatt egy adott problémához csak igen különböző hatóanyagok és célpontok interakciójáról érhető el adat, illetve az elérhető bioaktivitási adatok is heterogének és csak részlegesen állnak rendelkezésre.

A hatóanyag célpont interakció predikció célja a bioaktivitási adatok előrejelzése például, hogy köt-e adott fehérje adott molekulához. Ebben a feladatban is a fent említett nehézségek jelennek meg: a molekulák és a célpontok is heterogén adatok formájában állnak rendelkezésre és az adatok ad hoc, töredékes jellegűek, ahol a hiányzás is informatív. A heterogén többfeladatos bioaktivitás adatok nagyléptékű felhasználása máig is egy nyitott kérdés. A manapság egyre jelentősebb elosztott adatok optimális felhasználásánál remélt transzfer hatás felderítése érdekében idealizált többfeladatos tanulási szcenáriókat alakítottam ki és dolgoztam fel, ezen kívül egy protokollt készítettem, hogy maximalizálhassam az elérhető transzfer hatást.

A többfeladatos tanulás célja tipikusan a szélesebb körben használható, jobb általánosító képességű modellek előállítása, a gyógyszerkutatáson belül speciálisan nagy jelentőségű a több támadáspontú hatóanyagok kutatása. A többfeladatos jelleg jelenthet több kimenetet különböző szemantikával vagy különböző skálán, vagy a teljesítmény növelését több metrika szerint komplex veszteségfüggvényekkel. A leírt feladat többfeladatos aspektusa a bioaktivitás adatbeli különbségekből fakad, amely a modellek kimenetén is különbözőségeket eredményeznek.

A többfeladatos tanulásban elosztott adat felhasználásából fakadó transzfer hatás többféleképpen is elérhető: egy lehetséges megközelítés, ha a modellek kimenetére többfeladatos adatot teszünk, hogy általánosabb látens reprezentációkhoz jussunk. A predikciós feladatban ez azt jelenti, hogy a modellek kimenetét bővítjük több különböző és különbözően mért hatóanyag-célpont interakcióval. Az elvárás ekkor az, hogy ezzel a modellnek jobb lesz az általánosító képessége, és a modell egyes paramétereire tekinthetünk a hatóanyag molekula leíróinak és a fehérjék kötőhelyeinek egyfajta általános látens reprezentációjaként. A többfeladatos tanulás egy másik formája lenne, ha a többfeladatos kimeneti adatok egy részét a modell bemenetére tesszük, így maximalizáljuk az elméletben elérhető transzferhatást. Ezen módszerek közül mindkettő hasznos lehet gépi tanulási modellek fejlesztésére.

A dolgozatomban vizsgált feladatot a gyógyszerkutatás két egyre gyakorlatibb szcenáriójában is megvizsgáltam: Az első az elosztott adatok felhasználására alkalmazott federált tanulás többfeladatos kiterjesztésében. Ebben a szcenárióban gyógyszerkutatók egy kisebb csoportja működik együtt egy federált tanulási környezetben. Minden partner rendelkezik saját egyedi célokkal és a tanításhoz hozzájárul a saját adatával. Ennek a szcenáriónak a célja, hogy minden partner adatát a lehető legmegfelelőbben használjuk ki, a partnerek saját modelljeinek fejlesz-

téséhez. A második szcenárió a nagy mennyiségű nyilvános adat felhasználásával foglalkozik. Ebben a szcenárióban egy specializálódott gyógyszerkutató adatai csak egy kutatási irányból származnak és különösen érdekelt a több támadáspontú hatóanyagokban, másnéven olyan molekulákban, amelyek együttes teljes profilja illeszkedik a célpontok egy megadott halmazára.

A dolgozatban áttekintem a többfeladatos tanulás lehetséges megközelítéseit, ismertetem ezek előnyeit és hátrányait. Az első szcenárióban a federált környezetben elérhető legjobb teljesítményt vizsgálom, eltekintve a federált séma biztonsági aspektusától. A többfeladatos tanulás két formáját összehasonlítva igyekszem megbecsülni a transzferhatás lehetséges maximumát. Bemutatok módszereket a többlet adat olyan jellegű felhasználására, amely a legjobb vagy egyformán jó minden résztvevő számára. A második szcenárióban bemutatom a specializálódott kutató partner esetét, amely a nyilvános adatokat olyan módon használja fel, hogy az más területről vett korábbi mérések eredményeit kiaknázza.

Abstract

In the field of drug discovery, the development of a new drug or retargeting of an old one can be a daunting task, consuming large amounts of both time and money. Thus, in silico drug research is gaining more importance than ever before. The process is also supported by the vast amount of public data and knowledge, in addition to data accumulated by pharma companies while carrying out former experiments. During this process, two main difficulties emerge: heterogeneity and sparsity of the data, namely widely varying compounds and targets, and the highly incomplete bioactivity data.

Prediction of drug-target interaction bioactivity values is a fundamental task of in silico drug discovery, e.g., whether a given compound binds to a given target protein. This task also suffers from the aforementioned difficulties: data is available for heterogeneous compounds and specifically, for heterogeneous targets, i.e., tasks, while sparsity is also abundant. Currently, the large-scale reusability of heterogeneous multitask bioactivity data is still an open question. I designed and evaluated idealized multitask learning scenarios to characterize available transfer learning effects and constructed a protocol for designing multitask learning architectures to maximize practically realizable transfer learning effects.

In multitask learning we typically optimize for more than one goal with the aim of creating generally more applicable models or jointly optimizing multiple objectives, such as in multi-target drug discovery. This for example can mean multiple outputs with different semantics or scale or enhancing the performance in multiple metrics with complex loss functions. The multitask aspect of the given problem can be observed in the difference between the assays leading to differences in the outputs of models.

The effect of multitask learning can be observed in several ways: one could be to have models with multitask outputs in order to leverage the formation of a general latent representation. In the detailed task, this would mean that the outputs are the interactions of drugs with multiple different and differently measured assays. This could lead to a more generalized representation of drug-target interactions as a unified representation of compound fingerprints and binding site characteristics of protein targets. Another form of multitask learning would feed all available multitask output data on the input of the model, to maximize the theoretical transfer learning effect of these models. These both are legitimate ways of improving the performance of machine learning models.

My paper also investigates two currently active problems in pharmaceutical research: The first being the multitask nature of federated learning. In this scenario, a dozen pharma companies want to cooperate in a federated environment, where each of them contributes with their own data, and each of them has their own unique goals. This section aims to utilize the data of each partner as best as possible to enhance predictions of the personal models. The second scenario investigates the use of a large amount of publicly available data. In this scenario, a

specialty pharma only has assays from its main fields of research and is especially interested in multi-target drugs, i.e., in drugs with special profiles over the specific set of target proteins.

In the paper, I will overview approaches in multitask learning, and discuss the method's pros and cons. In the first application scenario, I will investigate the best achievable performance for multiple pharma partners in a federated setting, ignoring privacy issues. I will compare the two forms of multitask learning to bound the best achievable transfer learning gain. I will also present methods to exploit the additional data in a way that is best for each individual or equally good for every participant. In the second application scenario, I will outline the problem of a single pharma utilizing publicly available data, in such ways that it can evaluate and potentially utilize formerly used assays from different fields of drug research in its multi-target drug discovery.

Chapter 1

Introduction

In the field of drug discovery, the development of a new drug or retargeting of an old one can be a daunting task, requiring large amounts of both time and money. Thus, in silico drug research is gaining more importance than ever before. The process is also supported by the vast amount of public data and knowledge, in addition to private data accumulated by pharmaceutical companies.

One of the largest fields of in silico drug discovery is drug-target interaction (DTI) prediction. This process intends to predict whether a given drug or drug candidate has an effect, e.g., binds to a given protein. The biological background for this is the following: a drug is a chemical compound that brings about a physiological change in the human body when it is consumed, injected, or absorbed, a target is a part of a living organism to which drugs bind in order to bring physiological change [29]. After the compound bound to a target they react and this way, they cause changes in the organism. The aim of this process is usually to block the target from causing unwanted catalyzed chain reactions in the organism. To measure this one can construct assays, which are measurements that test different qualities of the given compound. The data is usually available as a matrix of assays.

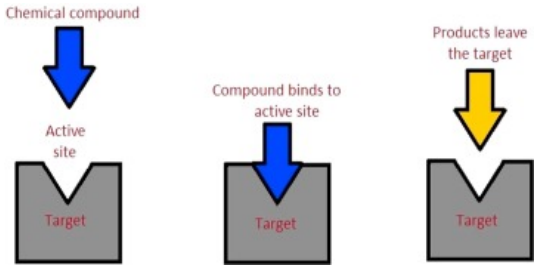

Figure 1.1: Drug-target interaction [29].

Traditionally to discover drug-target interactions, laboratory experiments are performed, but as said earlier these consume a large number of resources, thus computational methods emerged to reduce the search space for potentially working compound-target pairs. Currently, the use of machine learning models for this task is widespread. Neural networks are the most widespread and applicable models in these scenarios. They have achieved outstanding results in the field of drug discovery [26], and in my report, I will also investigate neural networks predicting such interactions.

The use of neural networks may be effective but presents unique challenges. One such challenge

is the multi-task nature of the learning process. This means that during learning the network optimizes for more than one target task, leading to a network that can simultaneously predict multiple outputs. However, the multi-task aspect of this problem is still unexplored, because of the rich repertoire of protein targets, their shared domains, and binding sites, and because of the heterogeneity of the measurements, as DTI data is accumulated from various participants of the pharmaceutical industry each using different techniques to carry out said measurements.

The amount of publicly available data in the field of DTI prediction is large and can take many forms. For my results, I used the ChEMBL database, which is a publicly available database of small molecules and biotherapeutics [7]. I adopted its binary version, which contains ones for active drug-target combinations and minus ones for the tested but inactive combinations. Because it is a missing-not-at-random (MNAR) [38] data, I used it with the following extension: all unknown value is marked as inactive, meaning that it was not tested either because it is most likely not a viable combination or simply because of the lack of resources. I used the Extended Connectivity Fingerprint (ECFP) [5] as descriptors for the compounds and predicted parts of the assay matrix.

The effects of multitask learning have been proven beneficial in the field of DTI prediction [24]. Multitask learning and transfer learning are best suited for deep learning approaches, as seen in computer vision [17, 12], because this way the outputs may gain information from deeper representations. Unfortunately, standard deep learning architectures have not yet emerged in drug discovery, thus, we have to settle for shallower models with enormous inputs [35].

The multitask challenge of the DTI problem is further escalated by the fragmentation of the DTI data, such as public and private DTI data sets, and private data sets of pharmaceutical partners. The joint use of public and private DTI data and the cooperation of multiple pharmaceutical partners is an open issue in the field of drug discovery [22]. This latter would mean that a dozen or more participants of the industry offer their data for the community and in exchange, they get data from everyone else to help in their unique goals. This leads to the world of distributed learning [16] and federated learning [14], where the fundamental life scientific and statistical question is learnability of this multi-task problem from distributed, horizontally and vertically separated data sets [37], specifically, the exploitation of potentially related targets as tasks shattered among the public and private data sets. This fusion problem is the focus of my research, specifically investigation of the existence and characterization of the transfer learning effect in multi-task distributed learning and the development of appropriate machine learning methods. Thus, privacy considerations, although of fundamental importance for private pharmaceutical partners, are not investigated in my report, especially, that one of my investigated application scenarios correspond to the joint use of public and single partner private data sets.

An idealized approach for the distributed data sets could be to accumulate every data about the compounds and the assays. This results in a hypothetical large matrix of assays that contain everyone's previous measurements, and a large descriptor of the numerous compounds used. The data constructed this way will still suffer from the problems of task (assay) heterogeneity and redundancy, i.e., from the unknown dependencies between the tasks (assays). Hopefully, these problems can be solved with the use of multitask learning. In this scenario the effect of multitask learning can be observed in several ways: one could be to have models with multi-task outputs to leverage the formation of a general latent representation. This would mean that the outputs are the interactions of drugs with multiple different and differently measured assays. This could lead to a more generalized representation of drug-target interactions as a unified representation of compound fingerprints and binding site characteristics of protein targets. Another form of multitask learning would feed all available multitask output data on the input of the model, to

maximize the theoretical transfer learning effect of these models. In the report, I prove that both are legitimate ways of improving the performance of machine learning models.

The use of vertically separated data sets also raises fundamental questions, as it is related to the question of using analogous tasks and problems in induction and problem-solving [8, 30]. One such question is, what is the best achievable result for the individual partners. Another one would be, how to run the computations in a way that gives each partner approximately results of quality and quantity in proportion to their contributions in data. As we can see these are often two separate and contradictory problems, hence they require separate solutions. In the report, I will examine both approaches in detail and I give computational solutions for both.

Another dichotomous and public form of distributed multitask learning would be, to have one pharmaceutical company, who is interested in complementing its private data with public data in its focused research on a given disease. For this task, it knows that it can utilize given binding sites on some specific proteins. This reduces the number of assays it is interested in. Experience shows that existing drugs can often be used for curing diseases outside of what they were intended to. This suggests the possibility that a pharmaceutical company can gain information from assays outside of its current field of research. A good example of this would be multi-target drugs, which experts its effect through multiple binding sites.

The difference is significant between the two scenarios. For example in the second, single pharmaceutical company, scenario there are virtually no obstacles for using all the available data on the input of the model, whilst in the world of federated learning, this is mostly an idealized possibility. Although the classical multitask scenario of having multiple different outputs may be better used on the distinguished data of a small number of partners, meaning two to four, rather than the large amount of public data, which often does not contribute to the correct representation, because there is less correlation between the targets. These questions will be explored in further detail in the report.

In multitask learning, the task, which is optimized for is called a main task, and tasks, which are supporting or improving the main task are helper tasks. When selecting viable helper tasks for a multi-task scenario, multiple aspects must be considered. We need a limited set of tasks, that improve the results as much as possible. Also, it is worth mentioning that some helper tasks may deteriorate the results, which must be averted when doing multi-task learning. The selection of these tasks is an open question, which I will call the Task Subset Selection (TSS) problem. As the name suggests it is analogous to the Feature Subset Selection problem in statistics and machine learning [15], which aims at selecting optimal features (attributes, independent variables). In the report, I will explore the parallels between the FSS and TSS problems, and suggest a novel method for selecting optimal helper tasks in the field of distributed multitask learning.

My main contributions summarized in the report are the following:

- Neural networks with task inputs. I extended a framework for multitask learning, where I can implement neural networks with tasks as input and measure the effect of these candidate helper tasks on performance.
- Multivariate characterization of transfer effect in multi-task learning. I extended the pairwise investigations of task dependencies by bounding contributions of tasks using neural networks with task inputs and by Gradient Boosting as a Feature Subset Selection over candidate helper tasks.
- Drug-target interaction prediction using adaptive task subset selection. I developed and implemented a novel drug-target interaction method, which combines a search algorithm

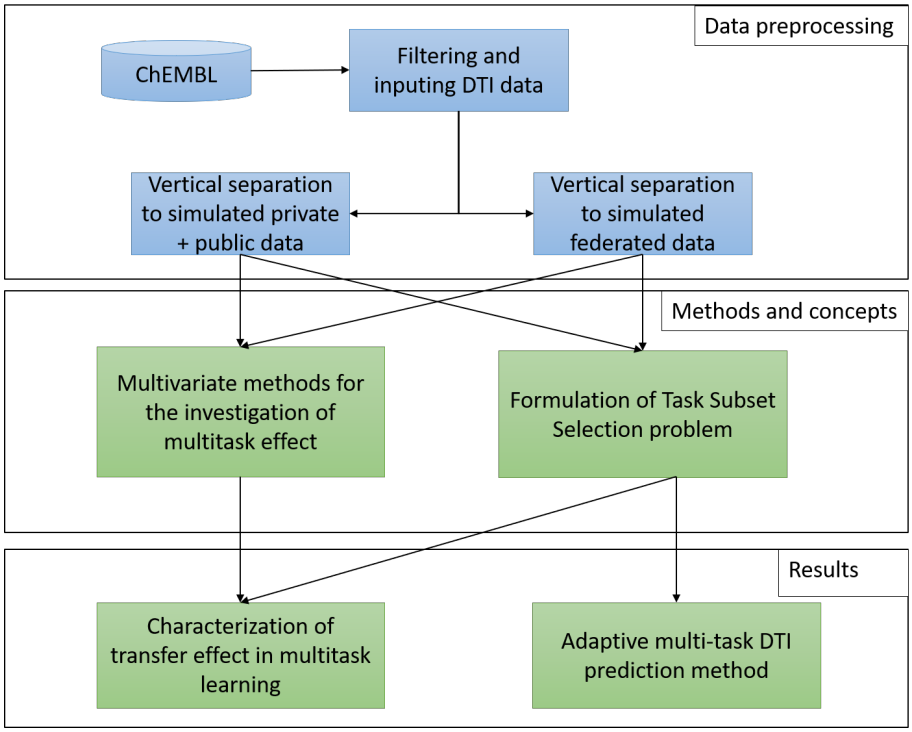

Figure 1.2: The outline of my work in the field of multitask learning. Blue representing data and green representing algorithms and experiments. The data comes from the ChEMBL database and preprocessed to get a simplified representation of bioactivity values. The data is partitioned to simulated pharmaceutical companies, and simulated public data is formed. First I developed methods to investigate the existence of the multitask effect in DTI prediction (green: top left). Next, I formulated the Task Subset Selection problem (TSS) (green: top right). Then, I characterize the multitask effect in the field of federated learning and the in the usage of public data (green: bottom left). Finally, I propose novel DTI prediction method exploiting the multitask effect (green: bottom right).

using Gradient Boosting, neural networks with task inputs, and multiple output neural networks to select appropriate tasks for multitask learning, i.e., to refine the structure of the predictive neural network model.

- Evaluations of realistic multi-party learning scenarios using real DTI data. I used the public ChEMBL data set to evaluate both the public-private and the multiple private partner scenarios.

The structure of this report is the following: First I will discuss multitask learning, its history, and advantages in the field of drug discovery. Next, I will present the applied data, introduce the concept of neural networks with task inputs to upper bound the transfer effect of tasks, describe the adoption of Gradient Boosting for the Task Subset Selection problem, and analyze the dependency structure of the tasks. In the main section, I will present two scenarios: a multi-partner cooperation, and a single partner using public data scenario, which represents a specialty pharmaceutical company focusing on multi-target drug development. Finally, I will compare and contrast the two scenarios and draw a conclusion on the work.

Chapter 2

Background

2.1 Multitask learning in neural networks

Multitask Learning is an approach to inductive transfer [3] that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better. Multitask learning may be applied in several machine learning algorithms so long as there are representations that can be shared between the tasks.

Multitask learning in neural networks the best way to understand the multi-task effect is to imagine several single task networks that are trained in parallel. The backpropagation is done for each task separately and each task develops the features in the hidden layers based solely on its own information. If the inputs of the networks are the same, it is easy to join them at some lower layer. By doing backprop for the new network the error is backpropagated through all the output tasks and in the next round, a task is able to use features developed by other tasks. If some developed feature would not be favorable for a specific task, it can always learn to ignore the hidden units responsible for that feature. This way the advantages far outweigh the disadvantages of multitask learning.

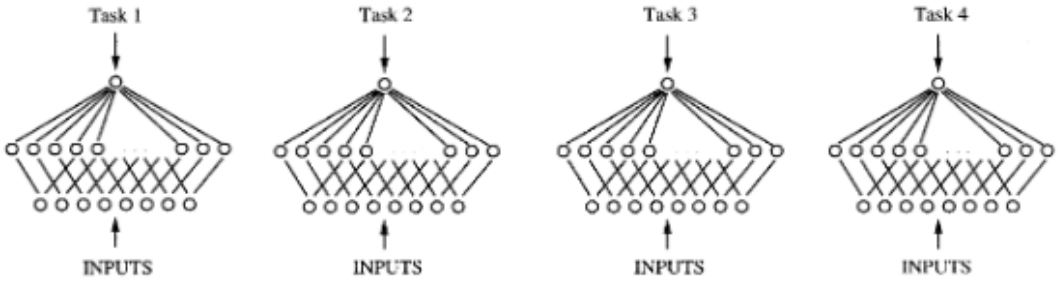

Figure 2.1: Single-task neural networks [3].

Multitask learning often appears in the form of a network learning the main task and having several other tasks to support that. The aim of this construction is as said before to make the supporter tasks provide bias for the main task. Bias in this context refers to the signals that make a model favor a hypothesis over another. This is crucial to the learning process.

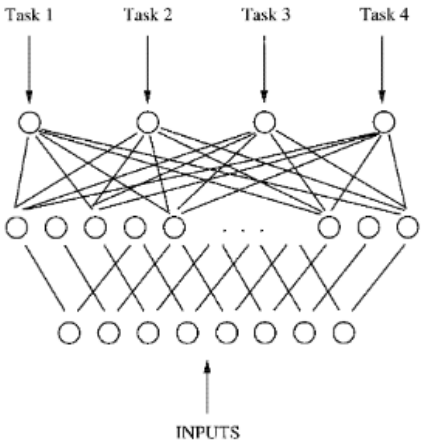

Figure 2.2: Multitask neural network [3].

The explanation for the benefits of multitask learning in traditional literature [3] are the following phenomena. Statistical data amplification is the ability of the network to distinguish the actual data from the noise added, this is helped by the representation that develops from the correlated tasks if we assume that the noise on the tasks is uncorrelated. Attribute selection is the process of selecting the correct input units for generating the representation, with one task the information for this is less, than it is for multiple tasks, thus the network will be more suitable to filter the relevant attributes to generate a representation. Eavesdropping happens when a representation is formed because of a task, and another task learns that it can improve itself from that representation, which may not have formed if we only trained on the later task. Finally, representational bias, which makes the outputs of the network converge in a common local minimum of the tasks if both tasks are likely to converge in it and makes them not converge in local minima that other tasks are not likely to converge in.

From another approach the multitask learning is a type of inductive transfer learning [40]. While classic transfer learning is applied when we have insufficient data from the target domain and we train to model on data from another domain and fine-tune it on the target data, thus we transfer information to the target domain. The multitask learning approach is usually used when there is sufficient training data available and the aim is to create more accurate or more generic models.

The main difference between classical knowledge transfer and multitask learning is that in multitask learning the learning of the tasks are not separated in time, while the process of knowledge transfer is usually used when a model was trained on one domain, and with knowledge transfer, it will be available to use in a different but related domain. In [33] they have proven, that both transfer and multitask learning is relevant to the field of drug discovery.

2.2 Multitask networks in drug discovery

Transfer learning has been present in the field of drug discovery since 2010. It was mainly used for small data sets [2], where training was not possible, thus related domain knowledge was transferred. It is still widely used for molecule generation, virtual screenings, and activity

prediction. Although classical transfer learning could improve the results on small data sets, the larger data sets draw more attention to the field of inductive transfer like multitask learning.

Multitask learning is a widespread method in the field of drug discovery. This is largely thanks to the large number of possible bioactivity that is useful and can be measured. The case of these values is typically helped with multitask learning as these multiple values on the output force a model to produce a more general representation of the given input molecules. The primary targets of these researches are quantitative structure-activity relationship (QSAR) tasks. DTI is a specific type of QSAR method, where we try to predict binary values for the otherwise quantitative task, in other words, DTI is a discretized QSAR task. However widespread they are the use of multi-task neural networks for QSAR tasks is a novel idea first deployed in the 2012 Kaggle Merck competition [13], where it earned first place.

A current study [40] attempted to explain the frequent superiority of multi-task models in the fields of QSAR prediction. Their initial thesis was that multitask learning performs better on smaller sets of training data while single task DNNs were better for predicting tasks with larger training data sets, however large variations occurred during the initial measurements. They devised experiments to measure the difference between the single- and multitask neural networks. And examined the best and the worst achieving multitask networks compared to their single-task counterparts. First of all, they found that similar input molecules play a huge role in the results of multitask learning. When the inputs are similar but the bioactivities are not correlated the multitask models performed worse than single-task models, however with similar molecules and correlated bioactivities the multitask networks were able to perform better. On different input molecules there seemed to be no difference between the single- and multitask approaches. Even a negative correlation of bioactivities helped because the network was able to learn the pattern of flipping the signs of weights.

It is worth to mention that the uncorrelated outputs produced a worse outcome, which seems to be in contrast with the previously mentioned research. The explanation for this may lie in the fact that these measurements were carried out on similar molecules, so the network should have given largely different outputs for molecules with little differences.

When the molecules that were active in the context of a given T task but were not active in another T' the, meaning the molecules were not similar the networks showed no positive nor negative improvement from its single-task counterpart. This is important because this way the joint modeling of tasks gets possible without any side effects, and computationally this will improve the performance of the model. This finding is mainly important to the modeling of dense data sets which rarely appears in realistic scenarios.

In another study [42] they successfully combine classical transfer learning with multitask learning to predict pharmacokinetic parameters of drugs. They did this in order to be able to learn from unbalanced data sets, which will be an extremely useful feature for federated learning multitask models.

The correlation aspect for the task is also analyzed in [24]. Which out of all the mentioned studies is the most similar to my approach as the number of assays a large and there are no designated main tasks, all are equally important. It is done on the same ChEMBL data set on which my results are computed, and here in vitro measurements are also conducted to verify the results.

A possible solution is outlined in [19], where they have compared different forms of multitask learning methods. They have devised a solution where they take smaller groups of tasks and perform a so-called multi partial multitask learning. This happens by clustering the targets and

doing the training on multiple networks with different clusters of targets to predict. They too have pointed out that doing multitask learning on not similar targets can have negative effects, which makes large-scale multi-task DTI prediction difficult.

A study conducted on multi-target drugs [20] multi-task learning is applied in a combination with target descriptors to identify possible new interactions for multi-target approaches. The solution also solves the inherent sparsity problem of DTI matrices.

In [4] the multi-task method is combined with a multi-view method, which helps in learning from multiple sources. The multi-view method is especially useful for handling public data which is oftentimes noisy or heterogeneous. This appears in the form of having multiple representations such as two different descriptors for compounds. The method of multitask learning described earlier (putting all available information on the input of the model) can be thought of as multi-view learning, where the descriptors of molecules are the traditional fingerprints used as input combined with the bioactivities of these molecules with different targets. The main difference being, that in the article separate predictions of views are averaged out while in my method the views determine one common outcome, and the weight of their importance is coded in the networks learned weights. Although the method is slightly different the principle is the same, that each view will get an equal chance to produce the best outcome.

In [27] the multi-task method is shown to be beneficial in the field of multi-target compound prediction. Multi-target compounds are molecules, which interact with multiple targets. The article examined the multitask method in a QSAR field for cancer research. I will use the benefit of multi-target drugs working well with multitask learning in the specialty pharma scenario.

As summarized in [31] and [34] multitask learning is applied more and more in the field of QSAR prediction, but there are limitations. They concluded that the claims of multitask learning benefiting from similarities in the target and the input space, but more research is needed in the area. The method is relevant for the simple benefit of not having to train multiple models for different tasks, but the real advantage of it is the improved performance compared to single-task models.

As we see multi-task learning can take many forms and can be applied in DTI predictions in numerous ways. In my research, I have separated the two distinct scenarios as mentioned before. In these fields, according to previous literature new results can be achieved. The federated scenario is interesting, because we have to calculate the similarities between partners' multiple tasks, and this way select a number of partners who might be able to help the given partner's targets. The challenge is amplified by the fact that if a partner finds the optimal partner(s) to cooperate with, the optimal partner(s) might not be interested in this, because it would have someone else as an optimal partner. This way an ideal solution has to optimize everyone's gains and expenses. The single pharma scenario is another type of challenge, here the single pharma has to use a large amount of available data to filter out relevant targets and beneficially use them.

2.3 Gradient Boosting in drug research

Ensemble models also have been gaining popularity in the field of multitask learning. In [11] a fusion of black-box models is shown with the models solving related tasks and knowledge transfer appearing between them.

Gradient Boosting is a powerful method in QSAR research. A Gradient Boosting algorithm

approximates the target function with a set of weak learners, in an additive form. The group of weak learners is called a committee [9] or ensemble. A weak classifier is some model, which is better than random guessing, but not as good as the final model needs to be. We train a sequence of these models and apply them in an additive fashion to get the committee's result. Gradient boosting has proven to give better prediction results on some data sets than random forests or even single task neural networks [32]. The aspect which gives Gradient Boosting an edge over the mentioned technologies is that t is computationally faster. I use this property of the model in my experiments. I give a method to utilize these types of models to enhance training with neural networks, solving the TTS problem.

Chapter 3

The dat set

3.1 The ChEMBL dat set

ChEMBL is a publicly available database of small molecules and biotherapeutics [7]. The data is added and modified manually based on scientific publications and journals. The information selected for the database is made up of two main parts: details of the tested compounds and the parameters of the assay conducted, which include the possible descriptors for the targets associated. In terms of the representations, ChEMBL strives to give uniform representations for every entity contained.

From the ChEMBL database, several data sets have been constructed for DTI prediction. For my computations, I used two versions of the data set. The two versions only differ in size, the structurally they are identical. The dat sets contain the descriptors of small molecules and a matrix containing the measured assays for each molecule.

3.2 The data structure

For the descriptors of the compounds, several possible candidates are present. Molecules are usually thought of as a graph with atoms as nodes and bonds as edges. For this reason in computer science, the most popular representations include graph traversal e.g. SMILES, which is a format that represents molecules with a string using Depth-first search. For my research, I used the Extended Connectivity Fingerprint (ECFP). Fingerprints are descriptors of molecules based on their structural properties [5]. ECFP fingerprints are circular topological fingerprints, which are excellent for structure-activity modelling. They are learned representations of molecules [31]. Fingerprints are generated by iterating over each atom of the molecule and noting the circular atom neighborhoods in a bit string, often the bit string is shortened to the list of indices of the 1 bits however I worked with the original full-length string as this can be beneficial in machine learning applications. Fingerprints fall in the category of hashed molecular representations [6] and are widely used, because of their flexibility in representation. The benefit is similar to the one-hot vector representations of labels, they are comparable to word2vec representations [39]. Among ECFPs we make a difference based on the diameter of the circular atom neighbourhoods. For my task, I used the ECFP_6 fingerprint, which means that the maximum diameter of the neighbourhoods is 6. In scientific research, a diameter of 6 is often used for activity learning, because it provides more structural information than a diameter of 4, which is usually used for

similarity searching. The result is a string of ones and zeros, which is 442672 bits in the first dat set and 32000 long in the second one.

The first dat set contains 295750 compounds and 2808 assays. The second one contains 482158 compounds and 3547 assays. The bioactivity data is detailed in a matrix, where each row corresponds to a molecule and each column corresponds to an assay. For a given compound-assay pair three values may be present in the matrix: 1, meaning the assay was measured with the given compound and it was active; -1, when it was measured but found inactive; and 0, when the assay was not measured for the given compound, either because it is most likely inactive or there was no capacity for conducting said measurement. This type of data is called Not Missing At Random (NMAR) because the missing (the 0 valued) measurements also carry information. The data was available in a sparse matrix format, which means only the indices of non-zero elements are given along with the non-zero element itself. The sparse representation saves on memory usage but needs special tools to handle. For this technical aspect, I used SciPy's sparse representation for computations.

For the scenario where multiple partners are present, I used a split version of the second dat set. The split version contains the same molecules and assays only less of them. The data is realistically distributed between the partners. It is important to note that the split is not balanced in any way that is why it is able to represent a real-world scenario. In federated learning, we can talk about horizontal or vertical splitting. Horizontal splitting means that partners have different rows of the target matrix, and vertical means they have different columns. In real federated learning scenarios usually, both splits are presents, leading to every partner having their own submatrix of the main matrix with occasional overlaps. In the federated scenario, this can lead to difficulties in representations.

3.3 Initial statistics

Analyzing the data can show some interesting properties about it, most of the initial experiments were conducted on the first dat set, using the whole matrix without splitting. The first experiment was a Principal Component Analysis (PCA) conducted for the matrix of bioactivities. As we can see on the scatterplot in Figure 3.1 the PCA shows that most of the data is split on two main axes, meaning that it has two main components, this can mean that the data may be predictable by smaller networks.

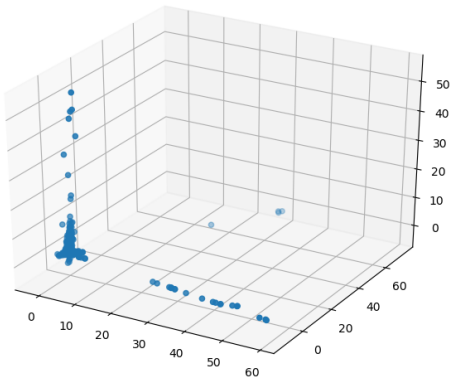

Figure 3.1: Principal component analysis for the bioactivity values.

After the PCA the next crucial knowledge is how similar are the outputs. This can be measured by the mutual information score. Unfortunately, when measuring the mutual information no real similarity was found between tasks, thus I had to look for other heuristics. As mentioned before the correlation between the tasks as random variables can be indicators of positive gain during multitask learning. Also mentioned before that the sign of the correlation is not important, because the neural network will be able to flip it. In Figure 3.2 we can see the absolute values of the correlations between assays. From the diagram, we can see that as expected there are tasks that often correlate with other tasks, and there are tasks that do not seem to correlate with any other task. These correlation values can be a useful heuristic when searching in the task space to maximize the transfer effect of multitask learning.

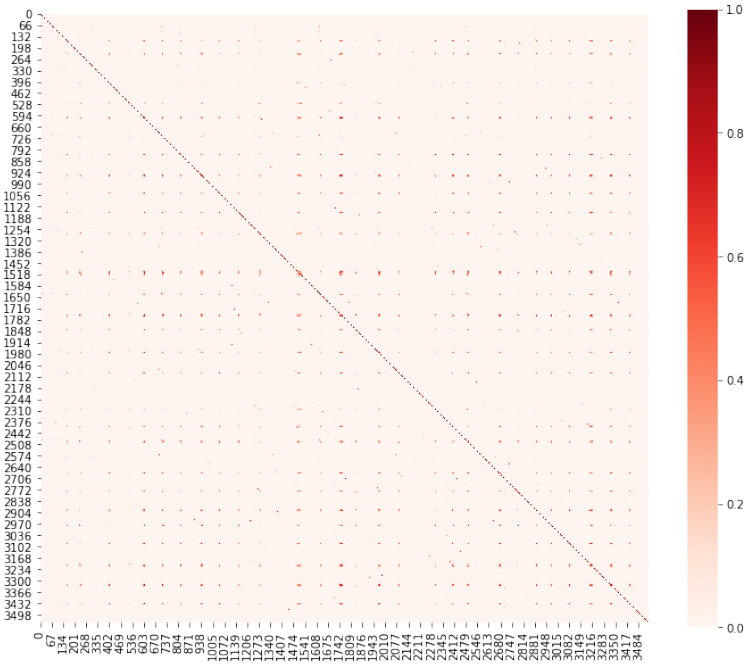

Figure 3.2: Absolute values of correlations for tasks.

At first, knowing the underlying structure of the data and the nature of multitask learning led me to the false conclusion that randomly selected combinations of tasks have a fairly high chance of improving the predictive performance compared to predicting each task one by one. I devised a set of experiments to test this hypothesis, which will be discussed in the following chapter.

Chapter 4

Methodology

4.1 The models

For predictive DTI models, the most widely used and successful are neural networks. This comes from their ability to build complex representations. I used a framework for building and training the networks developed at the Department of Measurement and Information Systems. The framework uses Python and PyTorch to handle models and can work with multiple models in a federated scenario. The framework is based on the Sparsechem framework [35], which has an open-source codebase. Sparsechem's goal is to support high dimensional machine learning models for learning sparse input biochemical tasks.

For certain experiments I needed more than just the networks which were already implemented in the framework, thus I added my own implementations to the framework. My contributions to the technical framework were networks, which had outputs in multiple layers, implementing losses to work on a per-task basis, meaning that the loss function only considers given outputs. I also implemented the searching algorithms, which search in the task space for optimal helper tasks, and added the possibility to train multiple models, on the same data, or train the same models by modifying the data.

For my networks, I used a feed-forward architecture. Most of my models have the same structure so they can be compared, when this is not the case I will mention it in the details of the experiment. The models contain two layers each containing 128 neurons. For activation functions between layers and for the outputs, I used a Rectified Linear Unit (ReLU). The ReLU on the output means that negative values are not predicted thus the output data is scaled for the evaluation to the range of [0-1], a 0 meaning inactive, a 0.5 meaning unknown, and 1 meaning active. For optimizer, I used Stochastic Gradient Descent with a learning rate of 1. The output loss was a categorical cross-entropy, adequate for a classification task. The data was split into five folds randomly and one of them was always used for evaluating the models, and the other four for training. I used a value of 0.2 as the dropout rate in every layer. Initially, I trained every network for 50 epochs, but these networks were overfitted. After trying multiple values for the number of epochs the most optimal was 20.

4.2 Pairwise multitask experiments

The initial hypothesis was that by combining random two tasks in a training the output will improve compared to training on one task. To test this first I selected ten tasks randomly and trained a model on them one by one, for 50 epochs. After the initial training, I selected two tasks to examine in detail, one of them achieved the highest loss after 50 epochs, meaning it was the least able to learn among the ten tasks and the other was the sixth in terms of lowest loss, meaning it achieved a medium result compared to the others. I will be referring to these two as main tasks.

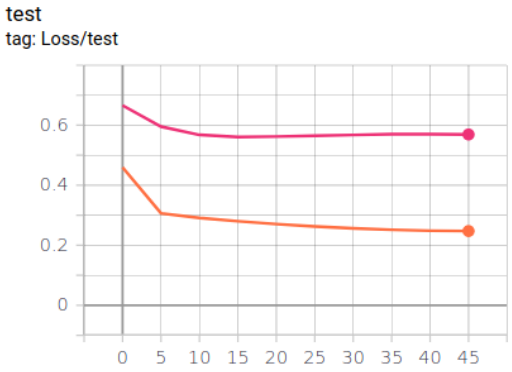

Figure 4.1: Initial training on the two selected tasks. (losses over epochs)

After the initial training sessions, I devised three experiments to measure the multi-task effects of the other 9 tasks used in training. In all of them, I had the main task combined with another from the remaining, I will call this helper task. The first is the classical multitask scenario, where the outputs of the model are one helper and one main task and they are computed in parallel. In this case, the loss of the model is the average of the two tasks, but the goal is to optimize the main task's loss. In the second scenario, the output layer is split into two distinct layers. The output for the helper task is calculated in a lower layer than the output of the main task, this way there is a more direct connection between the two tasks. The power of this model could be that the main task can simultaneously take information from the helper task, and the common representation of the two tasks. The third and final model was the one with the maximum information gain, here the helper task was directly fed on the input of the model, thus losing the benefit of the regularization from the common representation, but gaining exact information about the helper tasks.

As we can see in Figure 4.2 the training remained more or less the same no large changes can be observed in the diagrams. This may be due to the relatively low correlation values associated with the main and the helper tasks as discovered later. To ascertain that the multitask effect even exists in the context of the dat set I conducted a new measurement with all of the available tasks on the input as an ultimate solution with the maximum amount of information present on the input of the model. In Figure 4.3 the result is clear: the predictive performance has improved. The losses dropped considerably: from 0.57 to 0.34 and from 0.24 to 0.18. These numbers prove that the multi-task effect exists on the dat set.

Some questions do still remain: is this the best result that can be achieved on the tasks? I have previously mentioned, that multitask learning suffers from the fact that non-correlating tasks produce worse results than learning conducted on the tasks one by one, on the other hand since all helper tasks were on the input the network could have learned to ignore non-correlated tasks

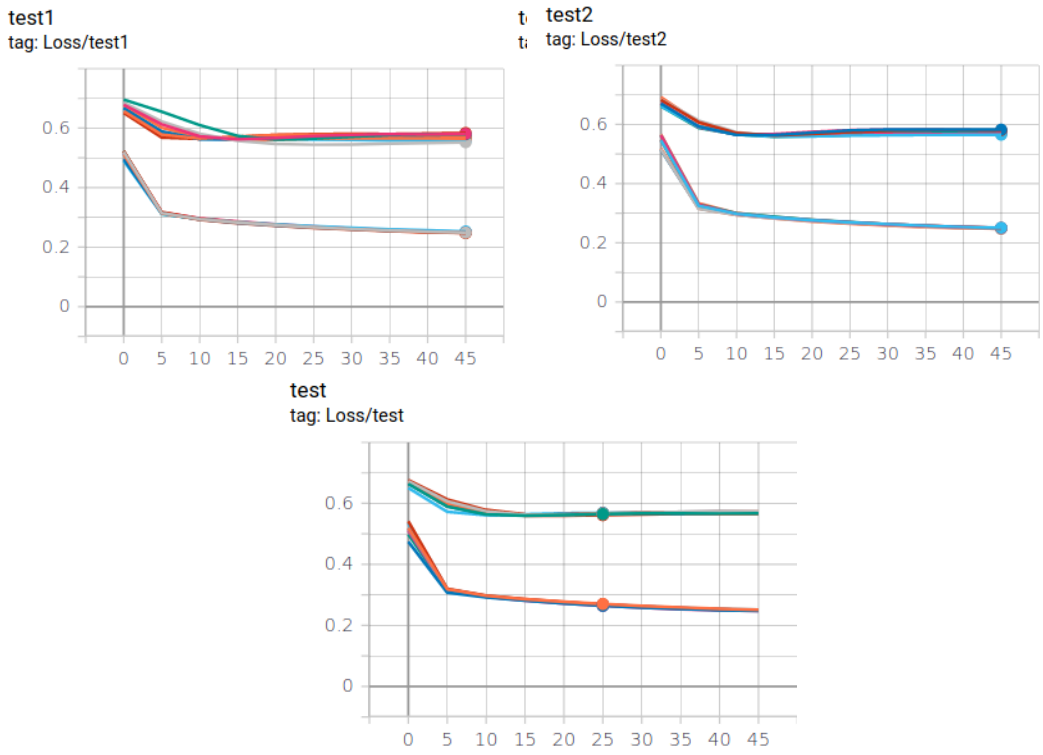

Figure 4.2: Results of the three experiments, ten training per main task on one diagram (First: upper left, Second: upper right, Third: lower. (losses over epochs)

to the main task. Another important question would be if the multitask effect can be observed in a similar scenario, but with all helper tasks on the output of the model, leading to a more realistic scenario. In the next two chapters, I will be investigating these and more questions relevant to the given scenarios.

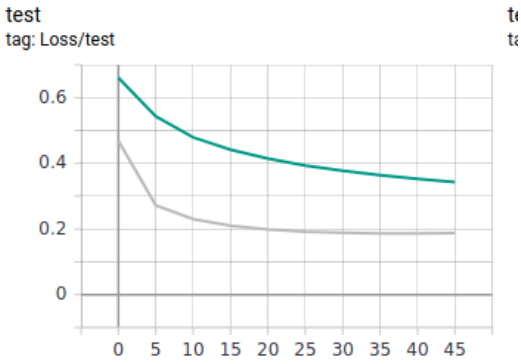

Figure 4.3: The main tasks on the output and all other tasks on the output. (losses over epochs)

4.3 Task Subset Selection

Selecting optimal helper tasks would be easiest if I could try and put each possible helper task on the input of a network and try to predict the main task, but it would be far too time-consuming

to do so. A good heuristic for this could be the correlation of main and helper tasks and since it is measured on the same molecules (when considering one partner) it will almost certainly produce better results, according to [40]. One problem would be to draw a line somewhere in the correlations and say with more or less certainty that only tasks above the line help the main task considerably. Another problem would be that tasks are only ever conditionally dependent on each other given the input fingerprint. For this, I would need a model that is capable of Feature Subset Selection. Luckily a number of such models exist one of them is a decision tree, which is able to select from the set of outputs that are the most relevant in terms of a given task.

Here I introduce the Task Subset Selection algorithm (TSS). It is a variation on Feature Subset Selection, applied in a multitask scenario. The goal of the algorithm is to select helper tasks for a designated main task in the context of multitask learning. For this purpose, I used a Gradient Boosted Decision Tree. For the purpose of this experiment the weak learners are decision trees, they have a maximum depth of three, which is consistent with the definition of the boosting algorithm. For this experiment I used scikit learn's implementation of the algorithm.

Chapter 5

Evaluation in multi-party learning

5.1 Federated learning

Federated learning (FL) is the process of multiple clients training the same model, while the training data remains decentralized [14]. This may or may not include a centralized server. It strives to reduce data collection and is able to overtake traditional methods in privacy and security aspects. Its most widespread form is cross-device learning, where a large number of clients with varying availability are present, a good example of this would be learning from private data on smartphones e.g. improving predictive text suggestions.

The other widespread branch is cross-silo learning, this usually means fewer clients with more data from one client. This almost never means more balanced data. In this chapter, I will examine this type of federated learning. As mentioned before the splitting of the data can be horizontal and vertical. I will use both horizontal and vertical splitting as this conforms to a real world application. The two main reasons for using cross-silo federated learning are the geographical separation of one company or the cooperation of multiple. The second one will be discussed in this chapter. The cooperation of pharmaceutical companies in drug research has gained popularity in recent years [25], this way they are able to innovate faster than the competition.

A different categorization of FL would be horizontal or vertical FL [41]. Horizontal FL is when the data overlap in the feature space, but not the id space, meaning the combined dat set contains more data points than each individual dat set, expanding the number of training samples. In vertical FL the id spaces overlap, but not the feature spaces, meaning the data is expanded in features, with the same number of samples as the individual dat sets. In this report, I will use vertical federated learning. Considering the number of models present numerous solutions exist [21], ranging from one central model to every partner having their own models. In this study I will use a different model for every partner, however, in real-world use cases, one-model approaches are dominant. Although my multiple model approach is easily generalized for multiple models as well.

Traditional challenges in FL are Data Imbalance, Missing Classes, Missing Features, Missing Values [1], these are an incomplete list, which does not always describe an FL scenario. In my scenario Data Imbalance is heavily present, because of the distribution of tasks, Missing Values are also a major concern while Missing Classes and Missing Features are not relevant to my scenario. Privacy is a central issue in FL, but it is not in the scope of this report, however, most

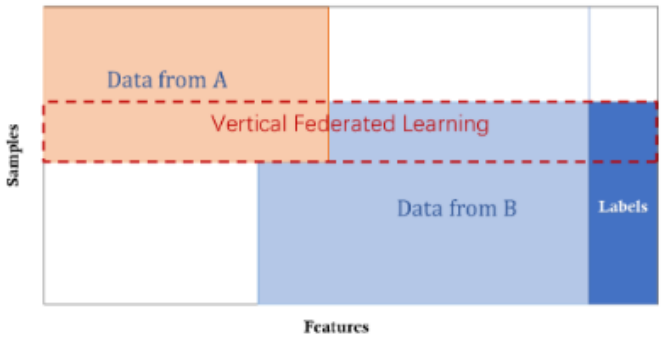

Figure 5.1: Vertical federated learning [41]

methodologies, especially neural networks can easily be implemented in federated settings with frameworks like [10] and [28].

5.2 The outline of the scenario

In this scenario, the goal was to get as realistic results as possible from public data. The number of clients, who hold the data will be ten. The second dat set will be distributed unevenly, but realistically between them both in terms of compounds and assays. But there are measurements for every assay of a partner combined with every compound. The number of compounds per partner ranges from around 5000 to more than 100000, the number of assays ranges from 2 to 340.

The goal of each partner is to improve the predictions for their own compounds, for this, they are willing to help others improve in their compounds. An important game-theoretical observation would be that they will only do this as long as their improvement is not considerably less than others'. In this chapter first I will try to improve the prediction scores for each partner as best as possible, and second, I will try to give a game-theoretical approach for a given partner about how much they would need to contribute to get the best results.

5.3 Pairwise cooperation of partners

First, it is reasonable to check if the partners were paired, how good a result can they achieve. This experiment gives an approximate result of how each partner contributes to the other's results. For this, at first, I train neural networks to predict all of the given partner's tasks, with a selected other partner on the input. I do this for all 90 combinations. This is the idealized scenario, so it will be complete if the calculations are done for the classical multitask version too.

Figure 5.2 shows, that by putting other partners' tasks on the input, we can improve the predictive capabilities of each partner's models. For the first two partners, there were no sufficient values in their tasks (probably because they have a small number of tasks), meaning, the AUC ROC values could not be calculated. The diagonal is empty because it would be meaningless to put the same task on the input as the output. The main observation was that for partners

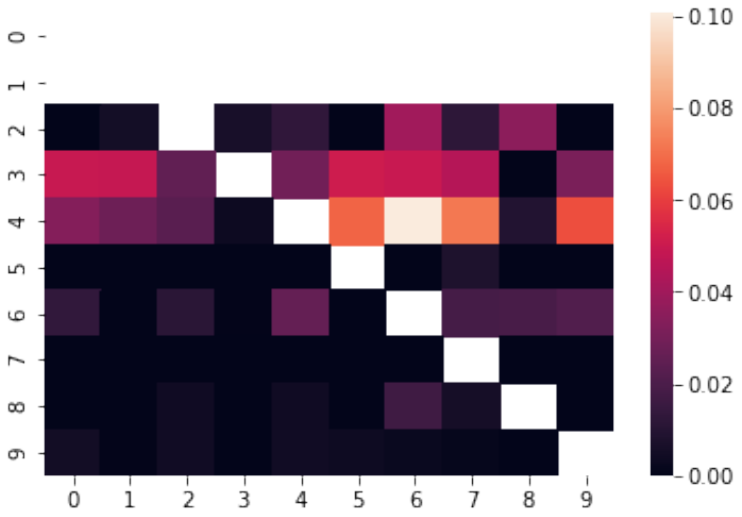

Figure 5.2: Improvement in AUC ROC value for every partner. (Idealized)

who have a relatively small number of tasks (partners 2-4), a much bigger improvement can be reached with extra information from other tasks. As opposed to partners with a larger number of tasks (partners 7-9), where almost no improvement was made by contributing tasks. Although for most of the partners some level of improvement was reached. For example, partner number 4 even had a 0.1 increase in their AUC ROC score, which is a considerable improvement.

5.4 Searching in one partner's tasks

If there is a goal to optimize for amongst the given partner's task it can act as a main task and all other tasks are helper tasks. It can be interesting for the partner that, which of their own tasks can contribute to predicting their main task. This experiment is a basis for some later experiments, as later we might be able to generalize it to searching in a space of partners or even their tasks.

The setup was the following: I selected one partner based on the number of tasks they owned, for this I used the partner with the largest number of tasks associated with them: partner number 8. The reason for this was that a task is only ever conditionally dependent on other tasks given the fingerprint. In practice, this means, that the fingerprint is always an input for the model, while the tasks are only if they are relevant. Thus the decision trees will select features mostly from the fingerprint and sometimes from the tasks as well. In the experiment it was crucial for the latter to happen, otherwise, it would be pointless to use helper tasks. And a large number of tasks on the input give them a higher probability that some of them to be relevant. In the end, the classifier was predicting an arbitrary task (T0) from the partner's set of tasks, and the other 339 tasks were on the input along with the fingerprint. The predictive capabilities of this model are not important in the long run, only the selected features matter, that is why the model can be trained relatively fast, which gives it an edge over a neural network. Fast training can be achieved with early stopping if there is no significant improvement.

In the end, the committee selected 25 features of the input as a base to classify the output, among which six were from the set of tasks. After finding the six tasks I validated the results with the correlation among tasks, the results show that from the six most correlating tasks four

overlaps with the selection of the classifier, and the remaining two selected also have a fairly high correlation with the main task. Figure 5.3 shows the correlation of all tasks of the partner by the number of correlation with the main tasks, the highlighted ones were selected by the TSS algorithm.

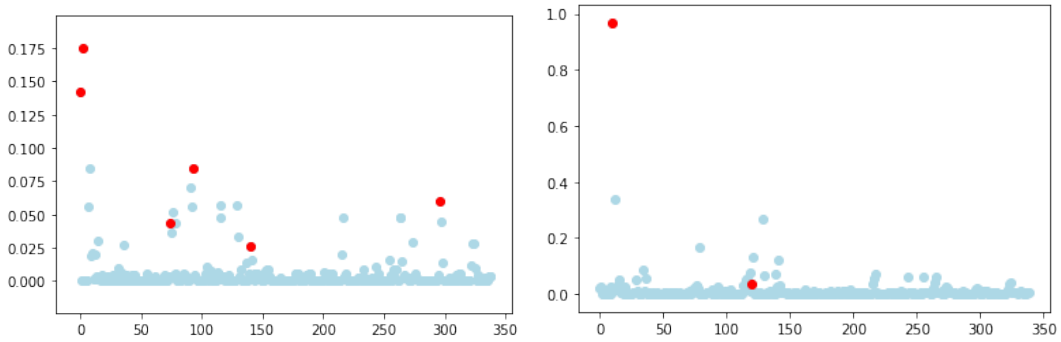

Figure 5.3: Correlations of tasks with T0 (left) and T11 (right), the highlighted were selected by the TSS.

The results seemed to be clear: the given tasks should improve the prediction accuracy for the main task. If the hypothesis is correct the neural network can be improved by the selected tasks on the input, and possibly the output as well. To test this I first trained the network with only the fingerprint on the input and the main task as output, then put each task one by one on first on the input. If the helper tasks are able to improve the network this way then this method is worth generalizing for full tasks in public dat sets or for selecting between other partners' tasks. Unfortunately, these tasks could not improve the capabilities of the model, although not even all of the partner's tasks were not able to do this. After this property was observed I picked a task where improvement was possible and repeated the experiment.

After finding a task that can be improved (T11) by others I repeated the experiment and here 84 features were selected from which two were helper tasks: the one with the highest correlation and one with a smaller value. After training a model with the help of the selected tasks almost the same level of improvement can be reached as if we had trained the model with all available tasks. As we can see in Figure 5.4 the task already produced good results, but in terms of loss, the helper tasks were able to improve the results considerably. As expected the bigger improvement came from the task which correlated more with the main task, when measured one by one.

Next, I tried to train a network to learn the tasks together in a classical multitask scenario. For this, the three tasks (the selected helpers and the main) together will form the output of the model and the fingerprint will be the input. The losses on the output are calculated together for training, but for evaluation, they are measured separately. As we can see from the losses in Figure 5.1 the classical multitask training is far better than the single task, but can not reach the information gain of the trainings when the helper tasks are on the input of the model. When considering the AUC ROC values for the main task after the training a similar pattern emerges. The single-class training was not able to reach an acceptable AUC ROC score, meaning, that it could not be used for prediction the classical multitask model falls between the no additional information and the maximal information case, but the classical multi-task's AUC ROC score is still not sufficient, only by having the helper tasks on the input can we reach a usable model.

To summarize by running the TSS a partner can get information about the underlying structure

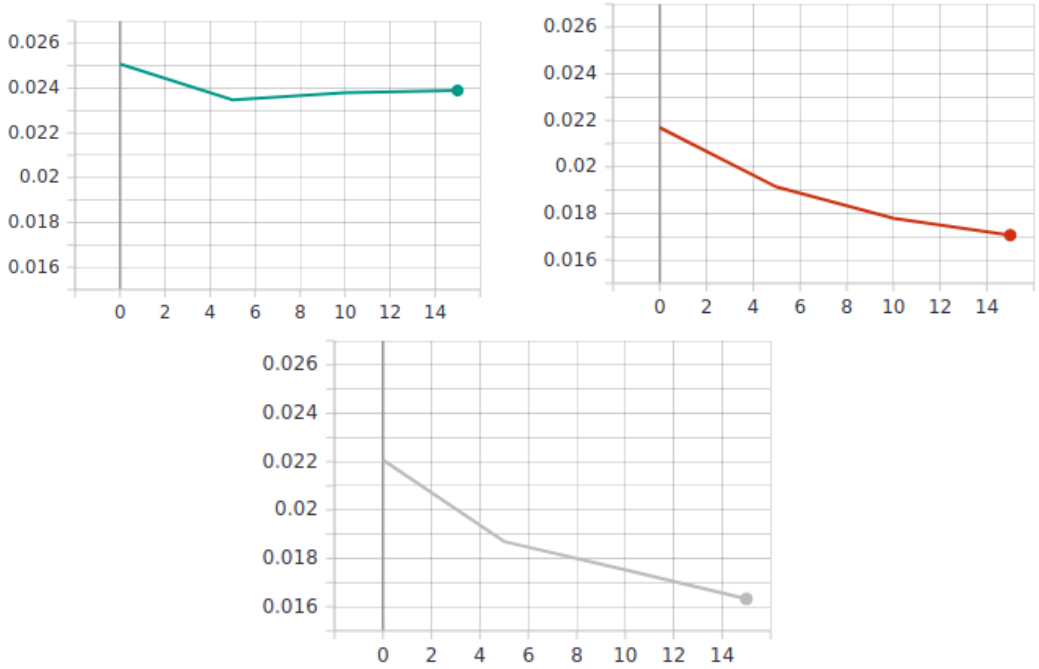

Figure 5.4: Loss on selected task, with no task on input (upper left), selected tasks on input (upper right), all tasks on input (lower)

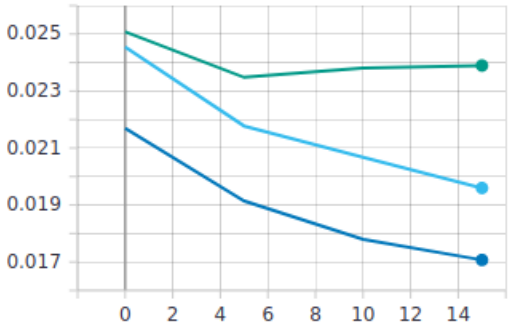

Figure 5.5: Losses during training for Single task (green), Classical multitask (light blue), Helper tasks on the input(dark blue).

Method	AUC ROC
Classical multitask	0.5009
Helper tasks on input	0.9323

Table 5.1: AUC ROC values after training networks with different methods, to learn form a partner"s own tasks.

of its tasks. By executing the algorithm for each task of a given partner we can construct a graph of dependencies for every partner. In this graph, tasks are represented as nodes and a directed edge runs from task T_i to task T_j if T_j is conditionally dependent on T_i given the fingerprint. Of course, this graph is only the approximation of these dependencies by the Decision tree, nevertheless, it is useful, which is proven by the experiments. This information may be used

when we run the same selection algorithm in the space of other partners' helper tasks.

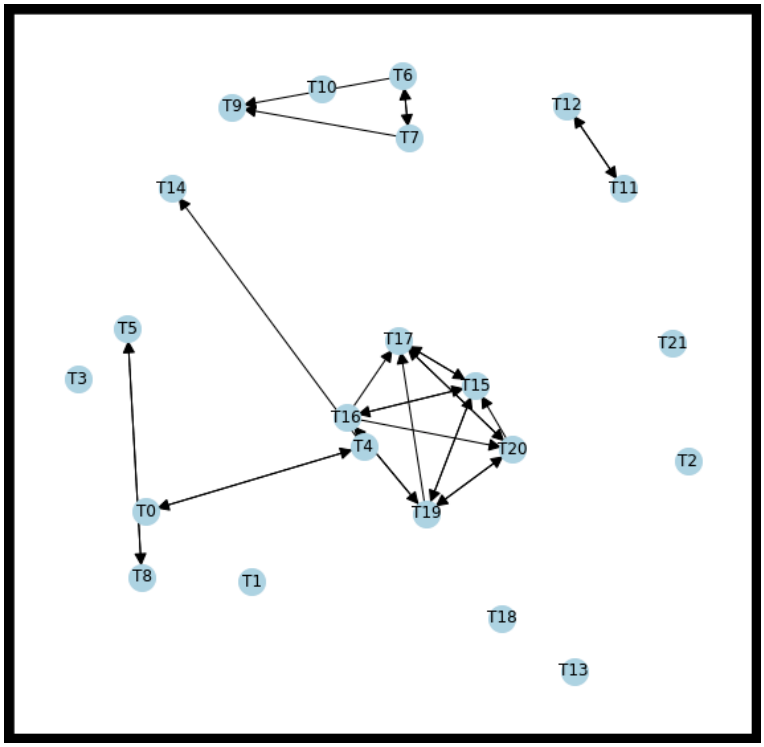

Figure 5.6: The map of dependencies in partner 4's tasks

5.5 Searching between other partners' tasks

After the previous experiment, we can see that the neural network can be improved by tasks selected from the TSS. Now I am going to make the first step in generalizing these results for the field of federated learning, so each partner can get the best results from the available data. For this, I will train the same ensemble learner with tasks from all partners. For this experiment, the assumption is that each partner is only interested in the molecules, which they own. The method will be the following: the tasks will be filtered to those, which have at least one overlapping result with the compounds that the selected partner owns. The input is formed the following way: all known assays will be concatenated with the corresponding compounds of the target partner, for compounds that have no measurement with the given target a zero is concatenated, in consistence with the NMAR principle. This data will serve as the input for the ensemble learning, and the neural network will be trained on the selected task, just like in the previous experiment.

For this experiment, I selected partner 9. For task number 278 of this partner, the TSS selected 41 features out of which 2 were a helper task. As we can see, the Decision Tree selected highly correlating tasks, just like in the previous case.

After the helper tasks were selected I trained a network to predict the task from the fingerprint only, another one to predict the task from the fingerprint, and the selected helper tasks another one to predict the task from all available tasks (which partner 9 does not own) and a last one to predict the values in a classical multitask manner (with the main and the selected helper tasks

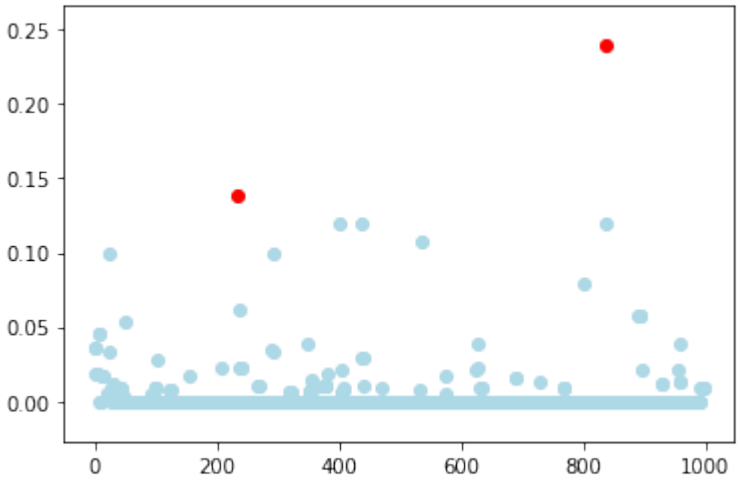

Figure 5.7: Correlations of tasks by a selected task for given partner, the highlighted were selected by the TSS.

on the output). Figure 5.8 illustrates the validation losses during training. In terms of AUC ROC value, the selected helper tasks perform the same as all tasks combined, and the model with the classical multitask scenario performs almost as good as the one with the selected task on the input, with all having at least 0.1 improvements to the single-task training.

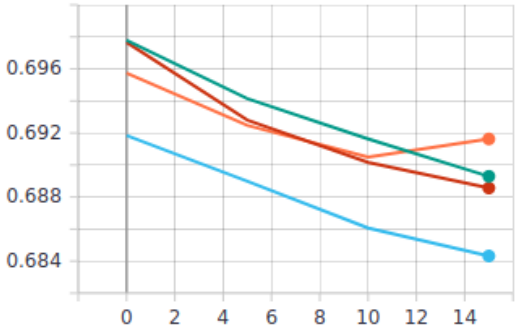

Figure 5.8: Validation losses during training for Single task (orange), Helper tasks on the input(red), All tasks contributed by other partners on the input (blue), Classical multitask (green).

5.6 Non-trivial usage: A simple metric for contribution

The previous experiment is useful in multiple ways: one is that it can show the relevant helper tasks for main tasks where improvement is possible, the other is that it can serve as a metric for measuring contributions. The metric can be constructed simply: each partner would select a set of its main tasks, after this we run a TSS on each of them, as this is a gradient-based problem, it can be calculated in a federated setup. After the algorithms are done each partner gets a score based that is calculated as follows: each selected main task is worth -1 point and each task that contributes to learning someone else's main task is worth +1 point. The sum of points will largely be greater than zero as for one main task multiple helper tasks exist. If someone's

score is negative or significantly lower they can be forced to contribute with more tasks or select fewer helper tasks. The variance in points indicates how equally beneficial the collaboration is. Based on these scores everyone can decide to stay in or drawback from the collaboration, and the group will see if someone benefits more and take action. A modified version of this can select the weight of each helper task (which is also determined by the decision tree) and assign a contribution score proportionately.

To visualize this metric a graph can be constructed similar to the one in Figure 5.6, the main difference being that each the dependencies are between different partners' tasks. This graph may be constructed for every partner aggregated or for every partner distinctly.

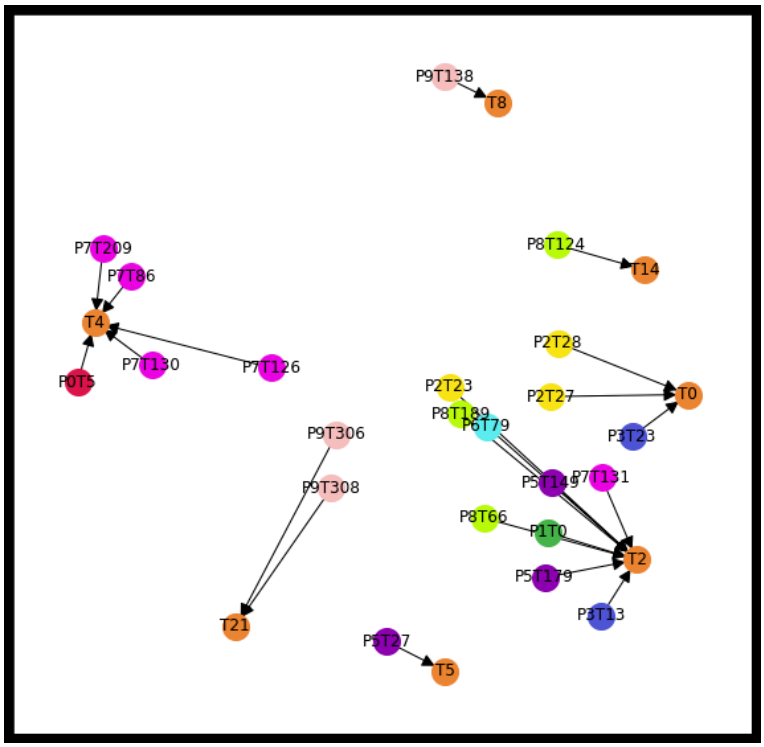

Figure 5.9: The map of dependencies between partner 4 and every other partner, each partner is represented with a different color (orange is partner 4), tasks are given in the format of P[partner]T[task], for partner 4 P4 is omitted

5.7 Results from the scenario

In this scenario, I showed how a group of pharmaceutical partners can cooperate in a DTI prediction task for drug research. I showed a method, with which partners can determine which task they would like to select as helpers for the learning of their own main tasks. I stated that the TSS is a viable method in this field and produced measurements that proved this. Finally, I showed an alternative use of the method, for scoring the partners by their contributions and gains, which can result in more balanced cooperation.

Chapter 6

Evaluation in the single-partner joint use of public and private data

6.1 Properties of a specialty pharmaceutical company

A specialty pharma is a development centric pharmaceutical company [18], mostly it is used in association with niche markets, where they develop compounds for certain diseases. The main distinction that makes a specialty pharma is that it is not a large capacity company, thus it needs special tools for development. Their business model is to acquire drugs from academia or contract organizations. The acquired drugs are then marketed towards specified physicians, this is why it can manage on a smaller level than a multinational pharmaceutical company.

Specialty pharma manages and produces specialty medicine. Specialty medicine cures conditions which are present in a small portion of the population. The proportion of specialty medicine approved increases yearly, meaning there is a high demand for these types of compounds. This process is improved by the breakthroughs in genomics and leads to enabling the market of personalized medicine.

6.2 Multi-target drugs

In the past, the one target-one drug paradigm was a leading paradigm, partly because it was thought to be effective to cure one disease with one drug, and partly to avoid unintended side-effects [23]. Yet new research on these often results in failure and on multiple previous one target drug's effects was proven to actually have affected multiple targets. Multi-target drugs are compounds that exert their effect through multiple targets. They usually do this by containing pharmacophores (or chemical features) with which they can bind to multiple target binding sites. The binding sites may be located close to each other or even on separate targets.

Multi-target drugs can be produced in numerous ways: either by combining existing single-target drugs or by joining existing molecules into one new molecule, or the simplest way is to identify some molecule that has multi-target properties, and identify its binding sites. The last method already conforms with the used method, this is why this property will be examined further in this chapter.

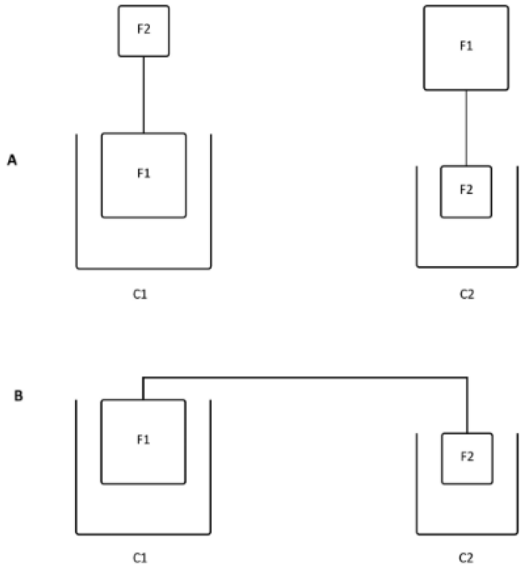

Figure 6.1: The binding of multi-target drugs to far away binding sites (A) and to close ones (B).

6.3 The outline of the scenario

In this scenario I will describe a specialty pharma, which wants to utilize publicly available data, to improve its drug development cycle. It will try to improve the research process by using data available about its compounds to identify new binding properties for them and analyze new targets to which the compounds can bind. Research has shown that models learning from public data can maintain their predictive power when applied to industrial tasks. [36] The aim would be to find multitarget properties of existing drugs, by analyzing other compound-target relationships.

For this scenario, I will use the same data as before only in a different context. I stated before that the split of data represents a real-world scenario, consequently, I will name some partners as the specialty pharma under examination, and other than their private values every data will constitute as public knowledge. I will improve DTI predictions for the analyzed pharma and try to identify multitarget drugs among its compounds.

6.4 Selecting helper tasks from public data

As we have seen it is possible to select tasks from domains with the same molecules as any given partner. In this section, the single pharma can be thought of as a partner from the federated scenario and by running the TSS for the publicly available tasks, it will be able to determine which tasks will help them predict new properties for the owned molecules.

For this, it can utilize the mapping of its own tasks and can construct a mapping of the helper tasks as well. The setup will be the following: we will consider only molecules owned by the pharma and try to predict the DTI values for them. For a maximum information case, I will put all available assays not owned by the pharma on the input of the model. For a classical multitask case I will put the same assays on the output of the model. In theory, the performance

of a single model should be better when, the tasks are on the input, as this gives more direct information.

The main difference between the scenario in the federated scenario and this one is that here all assays are available for the given set of compounds, not just the ones which were allocated to partners. The expectation is that the Decision Tree will select better helper tasks. This property of the scenarios is not always correct, in the real world data collected by big pharma is more reliable than publicly available data. For this reason among the experiments, I will include one, where data is mixed with noise, simulating a more realistic public dat set.

6.4.1 Selecting helper tasks for the input of the model

In this section, I run the TSS and select helper tasks for the given main task, as detailed above. I will then use the selected helper tasks to train a model with the selected helpers on the input. I will compare this model to the similar from the federated scenario. This scenario will also be examined on partner number 9's 278th task, thus I will be able to compare the results with the results from the FL scenario's 6th section.

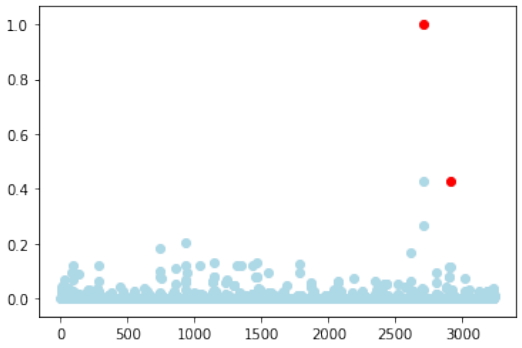

Figure 6.2: Correlation of partner 9's task number 278 with tasks from the public domain.

As we can see the correlations are higher than the ones in the previous chapter because more data is available. The Decision Tree selected the two most correlating task, with one of them essentially having the same values resulting in a correlation of one.

As expected the training with the helper tasks on the input gives a perfect result as opposed to the previous scenario. This is means that the selected helper task and the main task is essentially the same or same with a different sign, this explains the correlation of one. Of course, this leads to a perfect AUC ROC score.

Method	AUC ROC
Helper tasks from other partners on the input	0.5417
Helper tasks from public data on the input	1

Table 6.1: AUC ROC values after training networks with different methods, to learn form other partners' tasks (for partner 9's task 278).

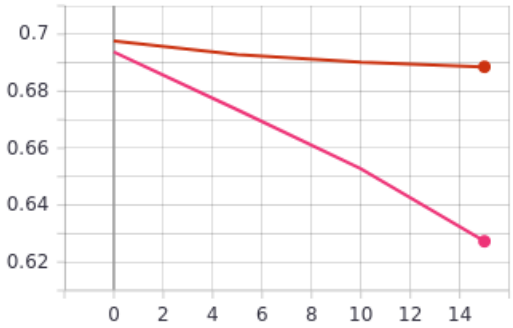

Figure 6.3: Validation losses during training for partner 9's task 278 with tasks on the input, tasks selected from other partners (red), and tasks selected from public data (pink).

6.4.2 Selecting helper tasks for classical multitask learning

This section will use the same main task and helper tasks as selected in the previous to train a classical multitask model with all relevant helper and main tasks as outputs and the fingerprints as inputs. I will compare the results with the previous section and the results achieved in the relevant section of the federated scenario.

As in previous multitask scenarios, I put the helper tasks on the output of the model. This method is worth comparing to the multitask case of the previous chapter's searching in other partners' tasks section and this chapter's previous section. Here the AUC ROC values, are fit for prediction, as opposed to the previous chapter's result. However, it is still not a strong model.

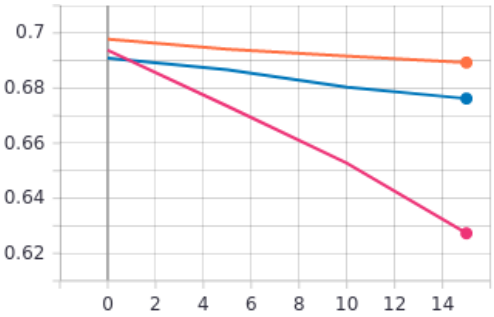

Figure 6.4: Validation losses during training for partner 9's task 278 with helper tasks, multitask case of the previous chapter form other partners (orange), tasks selected from public data on the input(pink), classical multitask case with tasks from public data (blue).

Method	AUC ROC
Classical multitask from other partners	0.4375
Classical multitask from public data	0.625
Helper tasks from public data on the input	1

Table 6.2: AUC ROC values after training networks with different methods, to learn form other partners' tasks (for partner 9's task 278).

6.4.3 Selecting helper tasks from noisy data

In this section, I will modify the data by adding noise to the public data of the scenario. In practice, this will mean changing the DTI values of the matrix with a given probability. I will compare these results with the results of the previous two sections, and measure the effect noise has on the dat set.

The data will be modified the following way: P_1, P_{-1}, P_0, will denote the probabilities of changing the ones, the zeros, and the minus ones respectively in the training data. This means for example that every zero of the dat set, which is not owned by the pharma is changed with a P_0 probability to 1 or -1. I will run the experiment with multiple values for the probabilities, and try to create realistic public data.

When the probabilities are too low there is not enough noise on the data, meaning the task, which had a correlation of one, was still close too similar, thus approximately $P_1 = P_{-1} = 0.005$ and $P_0 = 0.001$ are the smallest values where it is considered noise. When using these values the TSS selected 15 helper tasks. As the maximum correlation is lower the tasks selected are much less correlated than in the previous cases.

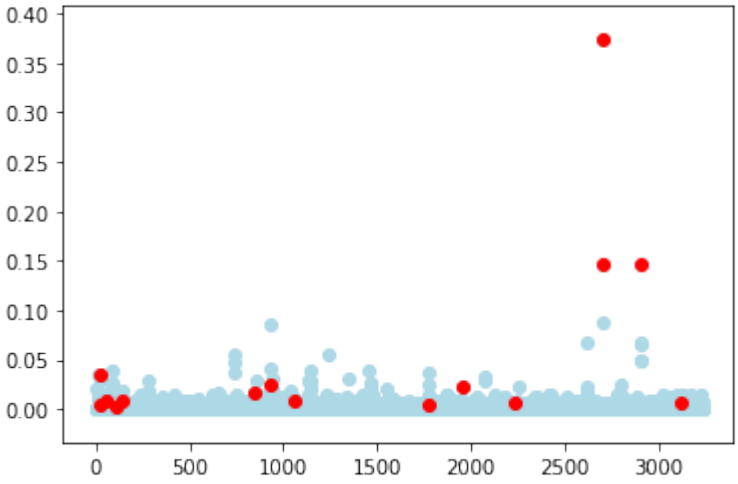

Figure 6.5: Correlations on the noisy data.

With the noisy data the network does not learn, the results fall back to the level, where they are comparable to the previous chapter's which is still impressive considering that random noise was added to the data, with relatively high probabilities for this large dat set.

6.5 Results from the scenario

In this scenario, I have demonstrated another field of use for the TTS algorithm. It was successfully able to identify correlating tasks, in a public dat set, to improve predictions on private data of the specialty pharmaceutical company. It was able to identify the multi-target drug in the dat set and predict their activities with the given targets. It also demonstrated its usefulness in noisy dat sets, which makes it a robust algorithm.

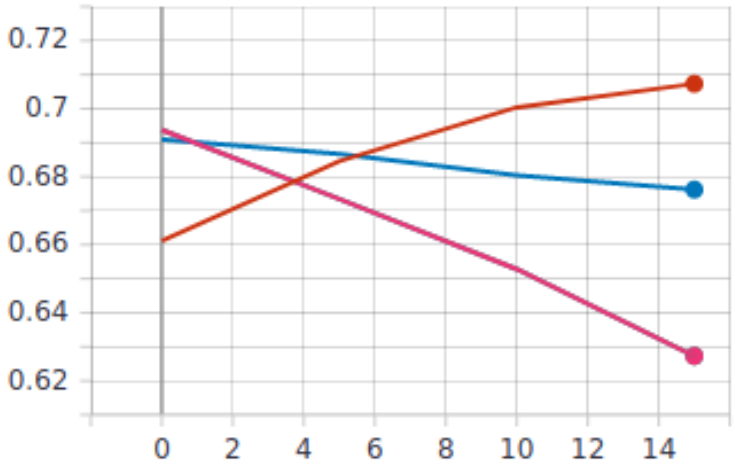

Figure 6.6: Validation losses of the previous methods (blue and pink) and the noisy data.

Chapter 7

Conclusion

In this report I investigated the effect multitask learning can have on DTI prediction. The initial hypothesis was that if we view targets as tasks, and try to predict their interactions with small molecules, data from other targets can help in the identification of these values. This hypothesis was confirmed twofold: for the case when we put helper data on the input of the models and for the case of classical multitask learning, when we predict the helper tasks with the task to be optimized. Although these cases have some restrictions. For one the improvement was only possible for tasks that the network was able to learn to some extent. There were tasks for which the network was not able to give good enough predictions, this was usually seen at validation, by the fluctuating validation loss. These networks were not helped even by selected helper tasks. The improvement was also shown in two fields: in the field of federated learning and the field of improving predictions from public data sets.

I have constructed a method of selecting helper tasks for any given main task in virtually any context. The TSS method uses the Gradient Boosting Algorithm and selects tasks that are conditionally dependent on the main task given the fingerprint. This approach is motivated by solutions for the Feature Subset Selection problem and is used in the context of improving multi-task learning scenarios with appropriate helper tasks.

In the federated learning scenario, I showed that every partner can improve the prediction of its main tasks by searching in their tasks space for suitable helper tasks. This way a partner can also construct a map of its own tasks and get a deeper understanding of their structures. After proving on a small scale that correlating relationships do exist in the domain of tasks, and these can be used beneficially, the next step was to search the space of tasks which were contributed by other partners. This search is based on a similar method, only here a larger domain is available, and there is less information available for the majority of the tasks, data is not owned by anyone, this results in helper tasks that are mostly zeros. Despite the larger domain and less information, improvement can be achieved in this field too.

In the single-partner scenario combining public and private data, I showed the benefits public data yields for a pharmaceutical company. I reconfigured the experiments in the federated scenario and tested them based on how applicable they are in this scenario. By not distributing the data to partners more data was available for training which is consistent with the real world amounts, with the exception that public data is more heterogeneous and less reliable. To model the drawbacks of public data I added noise to the data and showed a threshold of noise where multitask learning loses its benefit.

According to literature multitask learning for QSAR or DTI works well if the tasks are correlated

for similar molecules. As correlation is often hard to measure, especially in federated environments I used a simple learning algorithm to search for possible tasks to improve performance. The TSS algorithm is based on a Gradient Boosting Decision Tree, which was unable to predict the main task with the expected accuracy but was able to help select the candidates for helper tasks. As this algorithm is less reliable than a neural network it does not need as much time to complete and is also an especially well-performing tool for feature subset selection. The results of the algorithm almost always include tasks with high correlation to the main tasks as validated by computing the correlations.

The evaluations confirm that the method is viable and it scales well for larger data sets. By utilizing the algorithm federated learning scenarios can achieve a more accurate and/or more balanced cooperation. And the usefulness was also proven for non-federated applications. Although the method is useful there are still unexplored aspects of it. Future research may wish to explore a privacy-preserving version of the TSS algorithm for federated learning. The gap of performance between the classical multitask model and the one where the helper tasks are on the input is still considerably high, meaning that there is still room for improvement in the multitask models. As mentioned earlier generalization could also be made for a single central model federated learning scenario, by combining the selected main tasks and helper tasks in a wider model, this way the non-correlating tasks would not interfere with each other, and multitask learning could be conducted.

To summarize multitask learning for DTI prediction has proven to be a promising, but underutilized approach. In literature the multitask effect was analyzed and the underlying structure was ascertained to be the correlation of the tasks. In my report, I have tested these claims and demonstrated the widespread presence of multivariate dependency between the tasks, which should be taken into account in the development of efficient multitask DTI prediction methods. I suggested a conceptualization for this problem as the Task Subset Selection problem, and I developed a corresponding adaptive multitask selection DTI prediction method; adopting ideas from the feature subset selection problem and FSS approaches [15]. I tested the proposed method in different real-world scenarios and found it useful for most of them. I conclude that the method is fit for industrial use but still requires further research to optimize its performance.

Acknowledgements

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

List of Figures

List of Tables

Bibliography

[1] Mohammed Aledhari, Rehma Razzak, Reza M Parizi, and Fahad Saeed. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access*, 8:140699–140725, 2020.

[2] Chenjing Cai, Shiwei Wang, Youjun Xu, Weilin Zhang, Ke Tang, Qi Ouyang, Luhua Lai, and Jianfeng Pei. Transfer learning for drug discovery. *Journal of Medicinal Chemistry*, 63 (16):8683–8694, 2020.

[3] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[4] Sai Nivedita Chandrasekaran, Alexios Koutsoukas, and Jun Huan. Investigating multiview and multitask learning frameworks for predicting drug-disease associations. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 138–145, 2016.

[5] ChemAxon. Chemaxon documentation website. `https://docs.chemaxon.com/display/docs/Extended+Connectivity+Fingerprint+ECFP`.

[6] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(1):1–22, 2020.

[7] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[8] Pólya György. Indukció és analógia. a matematikai gondolkodás művészete, 1988.

[9] Trevor Hastie, Robert II Tibshirani, et al. The elements of statistical learning: data mining, inference, and prediction/by trevor hastie, robert tibshirani, jerome frieman. Technical report, 2009.

[10] Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[11] Trong Nghia Hoang, Chi Thanh Lam, Bryan Kian Hsiang Low, and Patrick Jaillet. Learning task-agnostic embedding of multiple black-box experts for multi-task model fusion. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[12] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

[13] Kaggle's blog. Deep learning how i did it: Merck 1st place interview. http://blog.kaggle.com/2012/11/01/deep-learning-how-i-did-it-merck-1st-place-interview/, 2012.

[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[15] Ron Kohavi, George H John, et al. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[16] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[17] Alex Krizhevsky and Ilya Sutskever. H. geoffrey e.,"alex net,". *Adv. Neural Inf. Process. Syst*, 25:1–9, 2012.

[18] Mannching Sherry Ku. Recent trends in specialty pharma business model. *journal of food and drug analysis*, 23(4):595–608, 2015.

[19] Kyoungyeul Lee and Dongsup Kim. In-silico molecular binding prediction for human drug targets using deep neural multi-task learning. *Genes*, 10(11):906, 2019.

[20] Limin Li, Xiao He, and Karsten Borgwardt. Multi-target drug repositioning by bipartite block-wise sparse multi-task learning. *BMC systems biology*, 12(4):85–97, 2018.

[21] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[22] Christopher A Lipinski, Nadia K Litterman, Christopher Southan, Antony J Williams, Alex M Clark, and Sean Ekins. Parallel worlds of public and commercial bioactive chemistry data: Miniperspective. *Journal of medicinal chemistry*, 58(5):2068–2076, 2015.

[23] Péter Mátyus. Több támadáspontú gyógyszerek: múlt, jelen és jövő= multi-targeting drugs: past, present and future. *Orvosi Hetilap*, 161(14):523–531, 2020.

[24] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24): 5441–5451, 2018.

[25] MELLODDY Project website. MELLODDY Project. https://www.melloddy.eu/.

[26] Ahmet Sureyya Rifaioglu, Tunca Doğan, Maria Jesus Martin, Rengul Cetin-Atalay, and Volkan Atalay. Deepred: automated protein function prediction with multi-task feed-forward deep neural networks. *Scientific reports*, 9(1):1–16, 2019.

[27] Lars Rosenbaum, Alexander Dörr, Matthias R Bauer, Frank M Boeckler, and Andreas Zell. Inferring multi-target qsar models with taxonomy-based multi-task learning. *Journal of cheminformatics*, 5(1):33, 2013.

[28] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[29] Kanica Sachdev and Manoj Kumar Gupta. A comprehensive review of feature based methods for drug target interaction prediction. *Journal of biomedical informatics*, 93:103159, 2019.

[30] Jürgen Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence*, pages 199–226. Springer, 2007.

[31] Jie Shen and Christos A Nicolaou. Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discovery Today: Technologies*, 2020.

[32] Robert P Sheridan, Wei Min Wang, Andy Liaw, Junshui Ma, and Eric M Gifford. Extreme gradient boosting as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 56(12):2353–2360, 2016.

[33] Rodolfo S Simoes, Vinicius G Maltarollo, Patricia R Oliveira, and Kathia M Honorio. Transfer and multi-task learning in qsar modeling: advances and challenges. *Frontiers in pharmacology*, 9:74, 2018.

[34] Sergey Sosnin, Mariia Vashurina, Michael Withnall, Pavel Karpov, Maxim Fedorov, and Igor V Tetko. A survey of multi-task learning methods in chemoinformatics. *Molecular informatics*, 38(4):1800108, 2019.

[35] Sparsechem code base. melloddy. `https://github.com/melloddy/SparseChem`.

[36] Noé Sturm, Andreas Mayr, Thanh Le Van, Vladimir Chupakhin, Hugo Ceulemans, Joerg Wegner, Jose-Felipe Golib-Dzib, Nina Jeliazkova, Yves Vandriessche, Stanislav Böhm, et al. Industry-scale application and evaluation of deep learning for drug target prediction. *Journal of Cheminformatics*, 12:1–13, 2020.

[37] Jaideep Vaidya. A survey of privacy-preserving methods across vertically partitioned data. In *Privacy-preserving data mining*, pages 337–358. Springer, 2008.

[38] Wikipedia. Missing data. `https://en.wikipedia.org/wiki/Missing_data#Missing_not_at_random`.

[39] Youjun Xu, Chenjing Cai, Shiwei Wang, Luhua Lai, and Jianfeng Pei. Efficient molecular encoders for virtual screening. *Drug Discovery Today: Technologies*, 2020.

[40] Yuting Xu, Junshui Ma, Andy Liaw, Robert P Sheridan, and Vladimir Svetnik. Demystifying multitask deep neural networks for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 57(10):2490–2504, 2017.

[41] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[42] Zhuyifan Ye, Yilong Yang, Xiaoshan Li, Dongsheng Cao, and Defang Ouyang. An integrated transfer learning and multitask learning approach for pharmacokinetic parameter prediction. *Molecular pharmaceutics*, 16(2):533–541, 2018.