CS230 Project Final
Using Deep Learning to Predict Toxicity and Lipophilicity from Molecular Fingerprints and 2D Structures

Riccardo Verzeni (rverzeni), Celia Xinuo Chen (xinuo)

June 9, 2019

Abstract

In Computer-Aided Drug Discovery, predicting molecular properties with conventional ML approaches (QSAR/QSPR[1]) heavily relies on domain specific knowledge. In order to overcome this limitation and provide a more general solution to this task, we investigated the use of Deep Learning (DL) to predict two very important properties for new drugs screening: toxicity and lipophilicity. We created from scratch different DL models: fully connected Neural Networks (NNs), consuming as input standard RDKit[1] molecular fingerprints and convolutional NNs (inspired by inception-resnet[2]) which directly use 2D molecular images as input. All different models produced promising results, especially in lipophilicity prediction, probably due to the large size of the available dataset. Furthermore all CNNs seemed to slightly outperform or in some cases at least perform as well as the fully connected NNs suggesting that the features learnt by the convolutional layers were sometimes better and sometimes at least comparable to those provided by the human engineered molecular fingerprints.

1 Introduction

Predicting molecular properties of drug-like molecules is crucial in new drug discovery. Studies show that with the recent raise of Deep Learning (DL) new interests arise for machine learning models which could perform on par with human expert predictions[3], suggesting an important renewed role for machine learning in Chemoinformatics. In this project, we will apply DL models to predict toxicity and lipophilicity, two very important properties for the screening and design of new drugs.

Traditionally, machine learning approaches such as QSAR/QSPR[4] heavily rely on domain specific knowledge for the features selection e.g. molecular descriptors and specific fingerprints. However, new research[3] suggests that we can feeds molecular skeletal formula images into deep learning models and let the model derive the relevant feature without explicitly including molecular descriptors for specific properties. This innovative approach could positively impact computer-aided drug design, for it could be easily re-purposed to predict a variety of different properties with minimal effort, making the prediction more efficient and less dependent on ad-hoc experimentally accumulated data.

For predicting toxicity and lipophilicity, we tried both: a more conventional approach and the above mentioned innovative approach. For the conventional approach we used molecular fingerprints as inputs, binary vectors of 2048 bits representing human encoded sub-structures of the molecule. For the innovative approach we used molecular skeletal formula images as inputs (2D arrays of pixels) Fig. 1. The outputs for the toxicity problem are binary values of whether one or multiple toxicological properties are present (represented as "1") or not (represented as "0") in a molecule. Instead the outputs for the lipophilicity problem are $\log P$ values (a measure of lipophilicity).

2 Related Work

Several papers explored how to predict molecular properties from the fingerprints. Pande et al.[5] use ECFP fingerprints as inputs, and construct a deep neural network using a multi-task, multi-layer perceptron architecture. Ramsundar et al.[6] explore how multi-task learning boost performance by predicting multiple molecular properties using the same fingerprint inputs and network. Some other papers experimented with predicting molecular properties from the molecular images. Mayr et al.[7] explain how to feed images into the deep neural network to predict toxicity, and Goh et al.[3] construct a residual neural network for classifying different toxicological properties. In addition, Bjerrum[8] gives a tutorial of how to preprocess SMILES into images to feed into neural networks. These papers were our points of reference when we explored our own neural networks.

3 Dataset and Features

We used two publicly available datasets: National Institutes of Health Tox21 dataset[9] and National Cancer Institute dataset[10]. The first dataset contains $\sim$ 10000 molecular structures along with values of some corresponding toxicological properties expressed in binary form (1: toxic, 0: non-toxic). From it we extracted $\sim$ 8000 data samples of molecular structures paired with the following toxicological properties: *nr-ahr, nr-ar, nr-ar-lbd, nr-aromatase, nr-er, nr-er-lbd, nr-ppar-gamma, sr-are, sr-atad55, sr-hse, sr-mmp, sr-p53*. Each data sample contains only a subset of the above mentioned toxicological properties. The second dataset contains $\sim$ 250000 molecular structures along with values of some corresponding physical properties among with $\log P$, a measurement of lipophilicity. From it we extracted two further subsets: 214648 data sample of molecular structures paired with presumably theoretically calculated $\log P$ values and 3576 data sample of molecular structures paired with experimentally measured $\log P$ values. All structures are initially stored in a structure-data file (.sdf)[11] which we then pre-processed. We use RDKit library[1] to parse the structure-data file and convert each structure into a SMILES[12] formula. They then get converted respectively into: (1) fingerprints, binary vectors of size 2048, where each element represents a manually encoded sub-structure of the molecule; and (2) skeletal formulas gray-scale images of size $150 \times 150 \times 1$, illustrated by Fig. 1, for the two different approaches respectively. Finally the binary vectors and pixel arrays are then stored with the respective labels/values into numpy ".npz" format files. The resolution of the images has been chosen to be high enough so that the atomic symbols in large molecules could be rendered correctly.

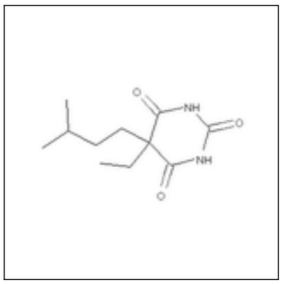

Figure 1: 2D image of skeletal formula sample from NCI dataset

4 Methods

We used fully connected neural networks with different number of hidden layers for the 1D RDKit[1] fingerprint input. We then use the convolutional neural networks (CNNs) for the 2D images of skeletal formulas with different architectures.

4.1 Architectures

The architecture of the fully connected neural networks are shown in fig. 2, 3

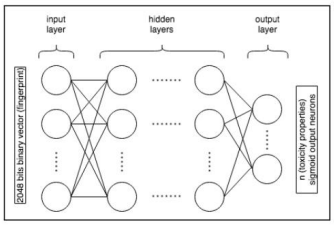

Figure 2: toxicity predictors fully connected model architectures

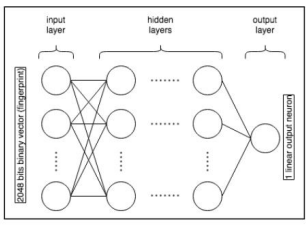

Figure 3: $logP$ predictor fully connected model architecture

In the different experiments we used different number of hidden layers and units (see the Experiments section for more details). For what concerns the CNNs we tried various approaches building up from very few layers and finally converging on the architecture shown in fig. 4 (called "incep_resnet_compact_v4" also referred as model-original in this paper). The convolutional model consists of 5 inception-residual blocks: the first four composed by an equal number of 3×3, 5×5 and 7×7 convolutions (type A) and the last one composed by an equal number of 1×1, 3×3 and 5×5 convolutions (type B). Each inception-residual block is composed by two inception blocks which double the number of filters but preserves the input width and height via padding. Each of the residual links are reshaped by a one by one convolution to match the number of filters. A LeakyReLU (rate: 0.03) activation is used after the addition of the shortcut. Each inception-residual block is followed by a 2×2 Max pooling layer that halves the input height and width. After the 5 inception-residual blocks a last 1×1 convolution layer is used to increase even more the number of filters before halving again the input height and width this time with an Average pooling layer. Finally the input gets flattened and fed to two fully connected layers with LeakyReLU (rate: 0.03) activation and Dropout (keep_prob: 0.75) before reaching the output layer. The output layer is respectively 1 linear unit for the $\log P$ prediction and n sigmoid units for n toxicological properties. Worth to mention that the architecture used for the toxicological experiments has been slightly different. Specifically it (called Model-m): (1) has simplified residual blocks for each (one concatenated layer instead of two); and (2) stacks more residual blocks together (a total of 12); (3) takes out the fully connected layers between flatting and output layer. For both fully connected and convolutional NNs the loss functions are:
Mean Square Error, for $\log P$ regression

$$\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \widehat{y}^{(i)})^2$$

Binary Cross Entropy, for toxicity classification

$$-\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log \widehat{y}^{(i)} + (1 - y^{(i)}) \log (1 - \widehat{y}^{(i)})$$

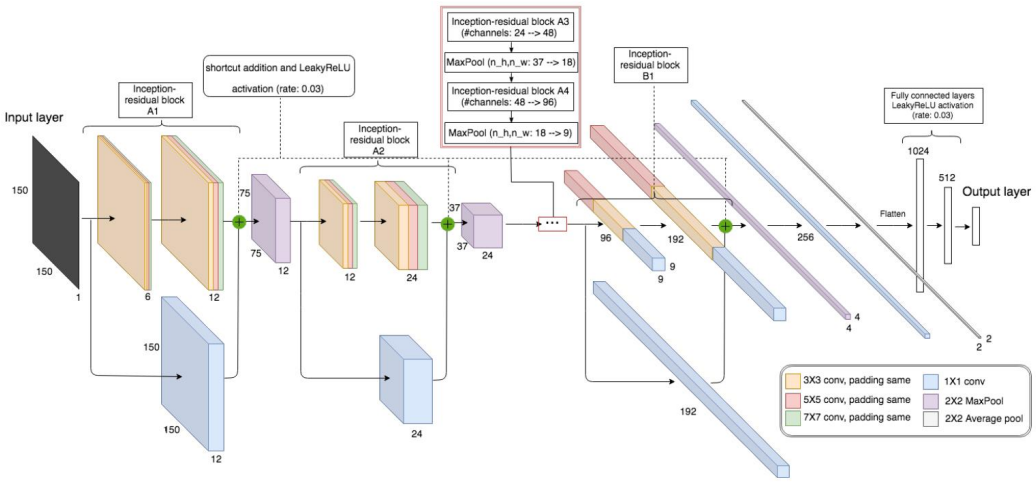

Figure 4: Model architecture of the best performing convolutional neural network inspired by Inception-ResNet [2] and Chemception [3].

4.2 Training

We used Keras APIs to build our models and train them. Respectively we trained the fully connected NNs on CPU while we used NVIDIA Tesla K80 GPUs on Google Cloud for the CNNs. All best performing models have been trained with mini-batch gradient descent (size: 32) using Adam optimizer and the respective above mentioned loss functions. Worth to mention that for the CNN with the largest dataset ($\log P$) we used 4 GPUs and a batch size of 256 in order to efficiently parallelize the calculations. Furthermore for deep CNNs model we used gradient clipping (0.5) and grad norm (1) for preventing gradient explosion and LeakyRelu (0.03) for preventing "Dying ReLU" problem. Finally for all experiments we used Keras ModelCheckpoint callback function saving the weights which best performed on the validation set effectively applying early stopping.

5 Experiments/Results/Discussion

Table 1: Result summary of experiments

	f1 score		Dataset size		
1.Toxicity models	Train	Test	Train	Val	Test
Image missing-label	0.892	0.487	7742	860	955
Fingerprint missing-label	0.839	0.495			
Image full input	0.911	0.456	6708	746	838
Fingerpint full input	0.864	0.428			
	R^2 value		Dataset size		
2. Liphophilicity (LogP) models	Train	Test	Train	Val	Test
fingerprints fcnn_6l_logP	0.966	0.839	173865	19318	21464
fingerprints fcnn_6l_experim_logP (transf learn)	0.985	0.923	2288	572	715
2Dimg incept_resnet_compact_v4_logP	0.963	0.852	173865	19318	21464
2Dimg incept_resnet_compact_v4_experim_logP (transf learn)	0.993	0.964	2288	572	715

5.1 Toxicity

5.1.1 Data split and evaluation

Within the Tox21 dataset, we picked out three toxicological properties with relatively complete data, *nr-ar, nr-er-lbd, sr-atad5*, and built multi-class classification neural networks that can use fingerprint inputs to classify all three properties at the same time. Based on the 8282 data we have, we set the train/validation/test set split as 81%/9%/10%. Since we have an imbalanced dataset where most molecules are classified as non-toxic and a few are classified with some toxicity, we use f1 score as the optimization metric, for a high f1 score means low false positives and negatives.

5.1.2 Multi-task Learning with fully connected networks

First, we experimented with fully connected networks that take fingerprint as inputs. We explored three kinds of fully connected network architectures (fig. 2): 2 layers (hidden layer: 2048 neurons), 3 layers (hidden layers: 2048, 1024 neurons), and 5 layers (hidden layers: 2048, 1024, 512, 256 neurons), all of which have an output layer of three neurons, each representing one of the three toxicological properties we are trying to predict. We then feed molecular fingerprints inputs with all three properties into each architecture, and see if the network could correctly classify all three at the same time. The 3-layer architecture achieved the best F1-Score of 0.495 on test set, followed by the 2-layer architecture, which achieved the F1-Score of 0.461.

5.1.3 Multi-task Learning with Convolutional neural networks

We moved on experimenting with convolutional neural networks that take images, instead of human-encoded fingerprints, as inputs, and see how much could engineer those features on its own. We fed our preprocessed image and true labels with/without missing labels in four models: (1) a 3-layer inception neural network; (2) the model-original described in section 4.1; (3) the model-m described in 4.1; and (4) the original ResNet 50 model[13]. We choose to base our experiments on residual networks, because it allows for deeper networks and gives better performance as the residual link could skip a layer using the identity function when no useful information is learned[3]. The result shows that model-m obtains the best result of 0.456 f1-score. fig.6 gives an activation map showing how the model is highlighting parts of the molecular structure and learning relevant features during training.

Table 2: confusion matrix of best performing model

	NR-AR	NR-ER-LBD	SR-ATAD5
True Positive	19	20	16
False Positive	7	22	11
True Negative	904	870	898
False Negative	25	43	30

5.1.4 Multi-task Learning with Missing Labels

A lot of molecular structures have missing labels for one or more toxicological properties. Because of that, we built a multi-task model that could train inputs with one or more missing labels, which should theoretically boost the performance by boosting the data size. To do so, we first preprocessed the dataset and fill in "-1"s for missing properties, and masked off the ones marked as "-1"s when calculating the binary crossentropy loss function, accuracy and f1 score. We then modified the above mentioned architectures to train with missing labels (which increased the data size by 15%). We found out that architectures with missing labels evidently

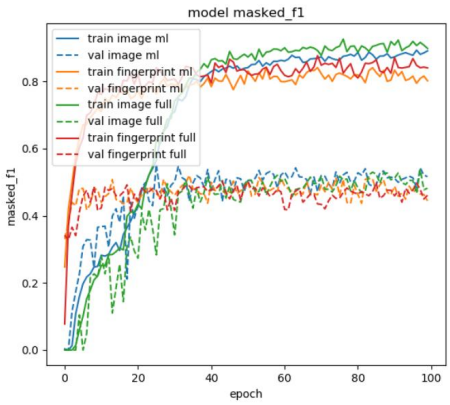

Figure 5: F1 score comparison for best-performing models using fingerprint/image inputs with missing labels/without

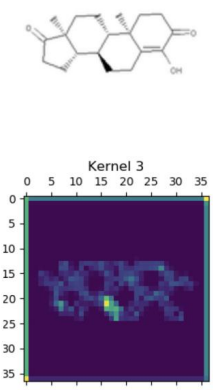

Figure 6: Activation map of the maxpooling (size 2 x 2), 16th layer of model-m

outperform those without. It brought up the result of the best-performing fully connected model from 0.428 f1-score to 0.495, and improved the result of best performing convolutional neural network from 0.456 to 0.487.

To compare the results we obtained from fingerprint and image inputs, fig. 5 illustrates the test and validation results, while table 1 gives a list of the test results. Overall, the performance of the best CNN is on par with that one of the best fully connected network, which uses fingerprint inputs. Considering the dataset has quite a small size and is highly imbalanced (1/9 toxic), both models have obtained reasonable results. Multi-label classification models with missing labels generally perform better than those without, most likely due to the boost of dataset size. Table 2 shows the confusion matrix of the best performing network (3-layer fully connected network).

Finally, we noticed that there is generally a variance issue, as most models attain high f1 score on training set (>0.8 f1 score) yet perform not as good on validation/test set (<0.6 f1 score). We tried to tackle it using a number of ways, using dropout(keep prob = 0.75)/l2 regularization for the fully connected network, and dropout/batch normalization for convolutional neural network. Unfortunately, none helped improved the performance in our case. We reasoned that the variance might be caused by complexity of the problem limited amount of data, so it could not be easily tackled by regularizations.

5.2 Lipophilicity

5.2.1 Data split and evaluation

We initially used the theoretical *logP* dataset (size: 214648) with a 81%/9%/10% (train/val/test) split. We then used the experimental *logP* dataset (size: 3576) with a 64%/16%/20% (train/val/test) split, for the transfer learning experiments. Given that the Lipophilicity prediction is a regression problem we used R^2 as evaluation metric and the mean of the true values as baseline (R^2 = 0).

5.2.2 Predicting *logP* from molecular fingerprints

For the fingerprint input we used a fully connected neural network fig. 3 experimenting 3 different architectures: 2 layers (hidden layer: 2048 neurons), 4 layers (hidden layers: 2048, 1024, 512 neurons) and 6 layers (hidden layers: 2048, 1024, 256, 512, 128 neurons). The 6L model was marginally better than the 4L one and both of them were reasonably better than the shallower 2L model (we omit the relative loss graphs included in the milestones for space constraint). All three models showed quite significant signs of overfitting so we then tried using L2 regularization with different values of λ and Dropout. Unfortunately although all the different regularization attempts reduced marginally the model variance all of them seemed to increase substantially the model bias leaving the un-regularized model still the best performing. All the loss, mean average error, and R^2 results on the test set of all trained predictors were slightly worse but very close to the ones obtained on the validation set. In Fig. 8 we can see the train and validation R^2 values during training of the 6L model (blue lines). The best predictor obtained a R^2 value of 0.839 on the test set, as reported in Table 1 (fingerprints fcnn_6l_logP) and Fig. 9, which is abundantly above the 0 baseline.

5.2.3 Predicting *logP* from 2D molecular images

We then moved to the 2D skeletal formula images inputs and tried to predict *logP* even without any human engineered feature, albeit general, which we used in the fingerprints approach. As mentioned in the Methods section we started with simple CNNs with few convolutional layers and then we moved to an inception CNN, then to an Inception-ResNet CNN progressively increasing the size of the network driving inspiration from Inception-Resnet[2]. Finally we doubled the number of inception blocks per residual block obtaining the model shown in Fig. 4. Increasing the number of convolutional layers progressively improved the results but also introduced two problems: "Dying ReLU" and Gradient Explosion. We noticed that over long training periods the network performances started quickly degrading suggesting a problem of "Dying ReLU". For that reason we switched from ReLU activation to LeakyReLU. That allowed the neurons to have a small gradients even for negative values. Secondly we noticed that even with LeakyReLU activation over long periods of training suddenly the gradient increased abruptly. We therefore applied gradient clipping (0.5) and force the gradient norm to be 1.

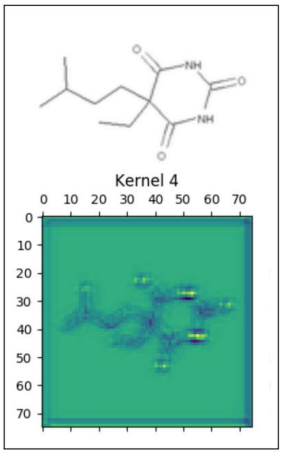

Figure 7: Activation map of the 5th filter of the 18th layer (5x5 convolution 8 filters) of model-original

At that point the model was robust enough to be trained for longer periods. We also attempted to use Batch Normalization on the convolutional layers but unfortunately the tuning of the momentum based on the batch size made it quite difficult to obtain good results, so we decided to not investigate it further. As seen in Fig. 8 (green lines) also the CNN shows some variance despite the use of dropout for the final fully connected layers but marginally better compared to the fingerprints fully connected model (blue lines). Comparing Fig. 9 and 10 we can see as on overall the CNN slightly outperformed the fingerprints fully connected model on the log P dataset obtaining a R^2 value of 0.852 on the test set reported in Table 1 (2Dimg incept_resnet_compact_v4_logP). As a qualitative measurement of the features learnt by the CNN $logP$ predictor you can see from the activation map in Fig. 7 that the model seems to pay particularly attention to the oxygen (O) and the nitrogen (N) atoms. Such atoms are quite polar which means that they affect significantly the way the molecule dissolves in water and lipids, ergo they are quite important for establishing the lipophilicity or log P of the molecule.

5.2.4 Transfer Learning Experiments

Since the size of the experimental log P dataset was quite small ($\sim$ 3500 data samples) we decided to use transfer learning picking the weights of the best log P predictors trained on the bigger log P dataset ($\sim$ 210000 data samples) and use them to initialize the weights of the experimental log P predictors for fully connected NN and CNN respectively. For the fully connected NN we tried freezing a different number of layers during training but without significant changes in the results obtained by not freezing any layer. For the CNN we froze all convolutional layers retraining only the final fully connected layers. In Fig. 8,9,10 we can see how both experimental log P predictors performed better on the small experimental log P dataset than the original log P predictors on the bigger log P dataset.

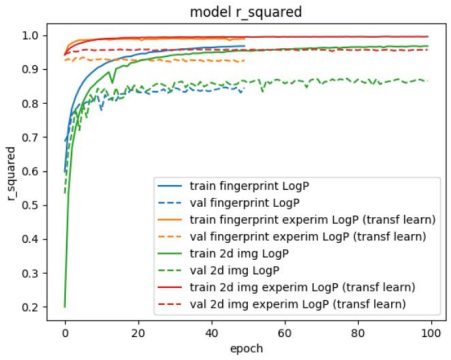

Figure 8: Comparison of all the best performing models R^2 during training. Fingerprints models have been trained for 50 epochs while the others for 100 epochs.

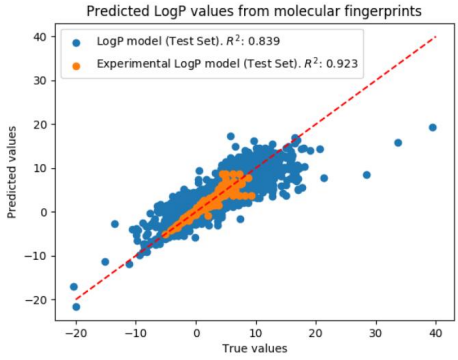

Figure 9: R^2 of LogP and experimental LogP from fingerprints. N.B: Experimental LogP model is trained using transfer learning from the LogP model.

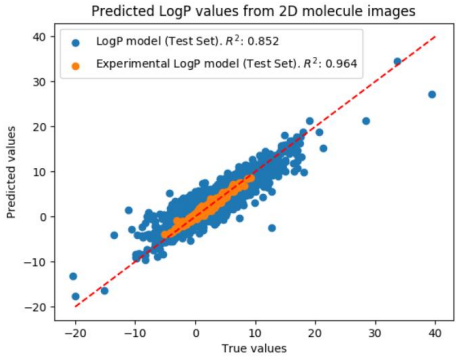

Figure 10: R^2 of LogP and experimental LogP from 2D molecule images. N.B: Experimental LogP model is trained using transfer learning from the LogP model.

6 Conclusion/Future Work

For the toxicity problem the best performing model is the 3-layer fully-connected network trained with missing labels (0.495 f1-score), followed by model-m residual network trained with missing labels (0.487 f1-score). Overall, the performance of convolutional neural network is on par with that of the fully connected network. Models trained with missing labels outperform the ones without, most likely due to the boost of data size.

While we achieved reasonable result for the toxicity problem, our results still have a noticeable gap with the best-performing literature [3]. It is likely because that we use relatively straightforward RDkit library to preprocess the fingerprint and molecular images, while a lot of high-performing models have more carefully engineered fingerprint inputs and image pre-processing like grid image[3]. Further, some models train with very innovative models, something we have not got the chance to explore. In future work, we would like to experiment with these different preprocessing and models. We also hope to gather more data to bring down the variance of the model.

The best LogP and Experimental LogP predictors are all abundantly above the baseline (R2: 0) and seems to predicts with reasonable precision the lipophilicity of the molecule. The transfer learning experiment seemed to boost the performances of the experimental LogP predictor, which outperformed the original LogP predictor with both inputs types. The CNN LogP and Experimental LogP predictors trained on the 2D molecule images slightly outperformed the counterpart fully connected neural networks trained on the molecular fingerprints suggesting that the features learnt by the convolutional layers were better than the human engineered ones in the molecular fingerprints. It would be helpful if we could gather more data to reduce the variance for the toxicity problem. We could also explore more state-of-the-art models and see if we could bring up the result to those of the best-performing papers. Given the encouraging results on predicting both LogP and experimental LogP value from 2D images it would be interesting to see if would be possible to improve even more the results with 3D molecular structure inputs and/or trying predicting different properties with regression.

7 Contributions

Both team members have equally contributed to the project. We have thoroughly discussed and collaborated on the literature review, model design, experiments and writeups. While Riccardo focuses on setting up the initial frameworks to both approaches and the experiments for lipophilicity, Celia concentrates on experiments for toxicity and substantial model adaptation for multi-task classification.

We want to thank our TA, Ahmadreza Momeni, for advising us throughout the project.

The code for the project is available at: https://github.com/project-cs230-2019/Project-CS230.git

References

[1] URL: https://www.rdkit.org/.

[2] S.; Vanhoucke V.; Alemi A. Szegedy, C.; Ioffe. Inception-v4, inception-resnet and the impact of residual connections on learning. 2016. arXiv:1602.07261.

[3] Charles Vishnu Abhinav O. Hodas Nathan Baker Nathan B. Goh, Garrett Siegel. Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. 2017. URL: https://arxiv.org/ftp/arxiv/papers/1706/1706.06689.pdf.

[4] URL: https://en.wikipedia.org/wiki/Quantitative_structure-activity_relationship.

[5] Evan N. Feinberg Joseph Gomes Caleb Geniesse Aneesh S. Pappu Karl Leswing Pande V. Zhenqin Wu, Bharath Ramsundar. Moleculenet: A benchmark for molecular machine learning. *arXiv:1703.00564*, 2017.

[6] Patrick Riley Dale Webster David Konerding Vijay Pande Bharath Ramsundar, Steven Kearnes. Massively multitask networks for drug discovery. 2015.

[7] Unterthiner Thomas Hochreiter Sepp Mayr Andreas, Klambauer Günter. Deeptox: Toxicity prediction using deep learning. *Frontiers in Environmental Science*, 2016.

[8] Esben Bjerrum. Learn how to teach your computer to "see" chemistry: Free chemception models with rdkit and keras. URL: https://www.wildcardconsulting.dk/learn-how-to-teach-your-computer-to-see-chemistry-free-chemception-models-with-rdkit-and-keras/.

[9] URL: https://tripod.nih.gov/tox21/challenge/data.jsp.

[10] URL: https://cactus.nci.nih.gov/download/nci/.

[11] URL: https://en.wikipedia.org/wiki/Chemical_table_file#SDF.

[12] URL: https://en.wikipedia.org/wiki/Simplified_molecular-input_line-entry_system.

[13] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. 2015.