Journal of Computer-Aided Molecular Design
https://doi.org/10.1007/s10822-022-00486-x

Enabling data-limited chemical bioactivity predictions through deep neural network transfer learning

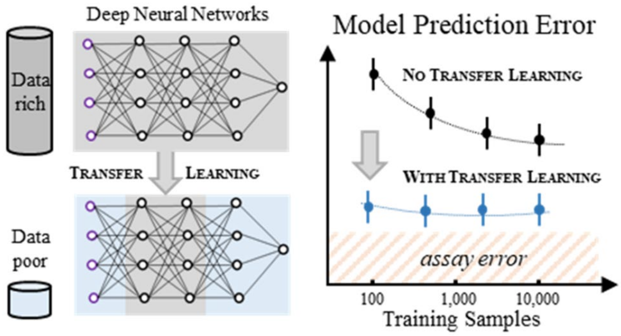
Ruifeng Liu[1,2] · Srinivas Laxminarayan[1,2] · Jaques Reifman[1] · Anders Wallqvist[1]

Received: 6 August 2022 / Accepted: 17 October 2022

Abstract

The main limitation in developing deep neural network (DNN) models to predict bioactivity properties of chemicals is the lack of sufficient assay data to train the network's classification layers. Focusing on feedforward DNNs that use atom- and bond-based structural fingerprints as input, we examined whether layers of a fully trained DNN based on large amounts of data to predict one property could be used to develop DNNs to predict other related or unrelated properties based on limited amounts of data. Hence, we assessed if and under what conditions the dense layers of a pre-trained DNN could be transferred and used for the development of another DNN associated with limited training data. We carried out a quantitative study employing more than 400 pairs of assay datasets, where we used fully trained layers from a large dataset to augment the training of a small dataset. We found that the higher the correlation *r* between two assay datasets, the more efficient the transfer learning is in reducing prediction errors associated with the smaller dataset DNN predictions. The reduction in mean squared prediction errors ranged from 10 to 20% for every 0.1 increase in r^2 between the datasets, with the bulk of the error reductions associated with transfers of the first dense layer. Transfer of other dense layers did not result in additional benefits, suggesting that deeper, dense layers conveyed more specialized and assay-specific information. Importantly, depending on the dataset correlation, training sample size could be reduced by up to tenfold without any loss of prediction accuracy.

Graphical abstract



Keywords Machine learning · Deep neural networks · Transfer learning · QSAR

Introduction

Numerous computational methodologies and applications have been developed to predict diverse properties of chemicals, such as their physiochemical characteristics, pharmacological effects, or biomedical activities [1–4]. The use of in silico modeling methods to develop quantitative structure–activity relationship (QSAR) play an important role in a

✉ Anders Wallqvist
sven.a.wallqvist.civ@health.mil

[1] Department of Defense Biotechnology High Performance Computing Software Applications Institute, Telemedicine and Advanced Technology Research Center, U.S. Army Medical Research and Development Command, FCMR-TT, 504 Scott Street, Fort Detrick, MD 21702-5012, USA

[2] The Henry M. Jackson Foundation for the Advancement of Military Medicine, Inc., Bethesda, MD, USA

Published online: 22 October 2022

range of disciplines such as rational drug discovery, toxicity predictions, and exposure risk assessments [5–8]. The fundamental steps in developing such model predictions require (1) acquiring training and validation data to build and evaluate the prediction model (data), (2) specifying a molecular description method to capture the relevant chemical features to build the model on (feature extraction), and (3) choosing a machine learning approach for modeling regression or classification (model) [9, 10].

Although these steps are interconnected, the main limiting factor in creating any accurate data driven machine-learning model is the availability of sufficient data to train the model, a problem that is especially acute when deploying deep neural network (DNN) techniques and models to QSAR-based chemical and bioactivity property predictions [11, 12]. This is in stark contrast to applications in image recognition where convolutional neural networks (CNNs) pre-trained on images belonging to a large number of readily available images can be re-used [13], effectively reducing the amount of training samples needed as there is no need to re-learn image feature extraction from scratch [14–16]. The ability to circumvent the need to re-learn the basics of image recognition is a powerful concept and, similarly, we would like to avoid re-learning all of chemistry and biology when developing machine-learning models that predict bioactivities of chemicals. Indeed, this transfer learning strategy was recently implemented and evaluated as a means to overcome the challenge of lack of data for deep learning in biomedical research [17–19], with an overview of deep transfer learning and related applications to drug discovery published in 2020 [20]. These studies show that deploying transfer learning to predict molecular bioactivity are likely to be successful, but the degree of success appeared to be dataset-dependent [18, 19]. None of the studies published so far systematically examined or determined the conditions under which transfer learning would be successful or quantified the potential benefits. Yamada et al. [19] examined a shotgun transfer learning method developed on 140,000 models pre-trained with large amount of training data. To developed a model with limited training data, they transferred a fraction of model parameters from one of their pre-trained models and optimized the rest of the model parameters using available limited training data. This process was repeated, each time with parameters transferred from a different pre-trained model. The resulting models were then evaluated, and the one that gave the best performance was chosen as the final transfer learnt model. This represents a trial and error approach and requires a certain amount of test data for a reliable evaluation of model performance, which may not be available if there is insufficient training data. Hence, the goal of our study was to develop and quantify a methodology that can be used to transfer learnt knowledge from one dataset to another prediction problem with a priori confidence.

Whereas image recognition techniques per se have not been widely adopted for bioactivity prediction problems [8, 21], the development of efficient molecular feature extraction methods can roughly be divided into a static, structure-based descriptor method that encodes atom and bond features [22–26] and a dynamic graph neural network (GNN) approach that learns molecular features within the context of the data to be modeled [27–30]. This latter method represents an efficient end-to-end deep learning method as the learnt, extracted features capture all features important to the modeled data, but may not be applicable for other datasets [31]. As a result, GNN-based transfer learning may instead lead to a degradation in model performance and learning [28]. Instead, here we are examining how and under what conditions the dense layers of one feedforward neural network can be used to augment training of a different DNN, i.e., transfer "knowledge" between networks, to classify a different bioactivity property.

The premise for this study is the application of the quantitative structure–activity relationship (QSAR) principle, which states that the chemical structure of a compound contains sufficient information to predict the outcome of a specific bioactivity assay. We fixed the representation of the chemical structures using extended-connectivity fingerprints (ECFP) to generate a one-dimensional fingerprint of each chemical structure—encoding atoms, bonds, and chemical environments—as molecular input features [26]. The choice of using ECFPs instead of graph neural networks (GNNs) to create molecular input features allowed us to focus on assessing the use of dense layers in applying transfer learning [32, 33]. We collected data for pairs of assays comprising one "large" and one "small" dataset, but with sufficient overlap between the chemicals to allow us to gauge the correlations between the two datasets. We then examined under what conditions (different network architectures, number of training samples, and degree of correlation between different assays) pre-trained layers from a "large" dataset could be used to develop a DNN model for a "small" dataset. In particular, we examined predictions of the 50% growth inhibition concentration (pGI_{50}) of compounds tested in the U.S. National Cancer Institute's NCI-60 Human Tumor Cell Lines Screening project [34]. These datasets comprise a varying number of overlapping compounds tested for their pGI_{50} in multiple cell lines [35]. We used these as well as other bioassay and physiochemical datasets to examine the reduction in mean squared prediction errors when transferring dense layers with frozen parameters from one fully trained DNN model based on a large dataset to train another DNN associated with fewer data.

The aim here is not to make accurate pGI_{50} or property predictions per se, but rather to examine under what conditions transfer learning is appropriate and what can be gained by transferring dense DNN layers. This paves the way for using similar transfer-learning techniques to overcome the challenge of limited data when implementing DNNs in drug design efforts, toxicity evaluations, and assessment of biological activity of chemicals.

Materials and methods

Molecular activity datasets

One of the challenges of evaluating deep learning is the lack of large high-quality datasets. Our study surveyed a total of 52 datasets with partially overlapping compound sets, ranging from a high of over 50,000 data points for the 50% growth inhibition of the A549 (a lung cancer) cell line to a low of 1266 data points for cytochrome P450 inhibition (Cyp2C9). The bulk of the data comprise 29 datasets from the U.S. National Cancer Institute's NCI-60 Human Tumor Cell Lines Screen project, examining growth-inhibition data for more than 60 human cancer cell lines of different tissues of origin for a large number of chemicals. For many cell lines, pGI_{50} spans 17 orders of magnitude, with a pGI_{50} uncertainty of about 0.45, estimated from multiple measurements of the same compounds that serve as plate controls. We also used 16 datasets ranging in size from 5923 to 2407 molecules each, comprising measured binding affinities of drug-like molecules to proteins as collected in the publicly available BindingDB database [36], and four datasets ranging from 3413 to 1266 molecules each, tested against different isoforms of cytochrome P450 inhibition from PubChem [37]. In addition, we used a publicly available acute rat oral toxicity dataset consisting of 6,320 tested chemicals [38] as well as a molecular lipophilicity dataset of 10,130 molecules, quantified by the n-octanol/water partition coefficient P and presented in a logarithmic form (logP), and an aqueous solubility dataset of 8,665 molecules, given in logarithmic form (logS) [39]. Details of these datasets, including the number of compounds in each dataset, molecular property/activity, measurement units, and source of the dataset, are summarized in Table S1 of the Supplementary Information.

DNN architecture, software, and hyperparameters

DNNs use a number of hyperparameters, some for defining the neural network architecture, such as number of hidden layers (depth of network) and number of hidden neurons in a hidden layer (width of layer), and others for controlling training behavior, such as gradient descent optimizer, learning rate, batch size, and number of epochs. We used networks with up to three hidden layers with different number of hidden nodes. For the single-hidden layer architecture, we used 100, 500, 1000, 2000, 4000, or 6000 hidden neurons. For the two-hidden layer architecture, we used all combinations of the first hidden layer containing 1000, 2000, or 4000 neurons and the second layer contained 100, 500, 1000, or 2000 neurons. We limited the three-hidden layer architecture to all combinations of 1000 or 2000 neurons in the first layer, 500 or 1000 in the second layer, and 100 or 500 neurons in the third layer.

To develop a best performing DNN for a specific dataset, a common practice is to first select a specific set of hyperparameters that works best for the dataset, which usually requires costly grid searches for large datasets [40]. As this procedure is dataset dependent, the optimal hyperparameters for one dataset may not be appropriate for another, and as transfer learning involves at least two different datasets (a data-rich and a data-limited dataset), we did not optimize hyperparameters for individual datasets. Instead, we used generally recommended or commonly used hyperparameters for molecular activity modelling [41, 42], and we deliberately constructed the above networks with different depths and widths to ensure that our findings regarding transfer learning were not dependent on a specific network architecture.

We performed all DNN calculations using the Keras API in TensorFlow 2.1.0 in a Python 3.7.6 environment. To prepare input data into training, validation, and test sets, we used the *train_test_split* function of scikit-learn 0.22.1. We used Adam optimizer [43] to minimize the mean squared error (MSE) loss function, with a learning rate of 0.001, a batch size of 50, and a maximum number of epochs of 2000. However, our DNN optimizations always stopped well before reaching the maximum number of epochs, as we applied the early stopping with a patience of 50, i.e., when the MSE of the validation set ceased to improve with 50 additional epochs, the training stopped and the network weights and biases that yielded the smallest validation MSE were selected as the final optimized model parameters. In an overwhelming majority of the cases, the smallest validation MSE were achieved in 10 to 30 epochs only. In order to reduce overfitting, we used dropout regularization with a fixed dropout rate of 25% on all input, output, and hidden layers. We used Keras' default selection/values for all other hyperparameters, i.e., the activation function was set to the ReLU function, the weight kernel initializer was set to "glorot_uniform" [44] and the bias initializer was set to "zero."

Input features

In this study, we used the counts of ECFP [26] features present in a molecule calculated using a bond diameter of two as the input features of the molecule. The fingerprint features were folded to a fixed length of 1024. That is, the input features of each molecule were stored in a vector of 1024 integers, where each integer represents a count of a molecular fragment present in the molecule. We generated the input features using Pipeline Pilot software using the Molecular Fingerprints and Convert Fingerprint components (Dassault Systèmes, Vélizy-Villacoublay, France).

Transfer-learning efficiency

To assess transfer learning quantitatively, we defined transfer-learning efficiency (TLE) as the percentage reduction of MSE due to transfer learning, as follows:

$$\mathrm{TLE} = \frac{\mathrm{MSE}_0 - \mathrm{MSE}_{\mathrm{TL}}}{\mathrm{MSE}_0} \times 100\% \qquad (1)$$

where MSE_0 represents the MSE of test-set compounds by a model trained without transfer learning and $\mathrm{MSE}_{\mathrm{TL}}$ denotes the MSE of test-set compounds by a model trained with transfer learning.

Results and discussion

Impact of neural network architecture and training set size

To evaluate the effect of network architecture and training set size, we used the largest NCI-60 dataset—the growth inhibition data of 50,606 compounds against A549 cell line. We randomly selected 1000 compounds as the test set for DNN model performance evaluation. From the remaining compounds, we randomly selected subsets, ranging in number from 500 to 40,000, as the training sets and used the leftover compounds as validation sets. Although this introduced variability of the size of the validation set used to determine the training stopping point, the final test set used to evaluate the results remained fixed for these calculations.

For single-hidden layer networks, we evaluated networks of different width, with the number of hidden neurons ranging from 100 to 6000. For each network architecture and training-set size, we repeated model optimization five times using different initializing conditions, with the average MSE of the test set from the five resulting models considered as the MSE of the network architecture. Table 1 summarizes the numerical evaluation as a function of hidden neurons and number of compounds in the training sets, showing that with a large number of training samples (i.e., ≥ 10,000), the number of hidden neurons did not have an impact on model performance. However, when the training set was smaller, models with too few (i.e., 100) or too many (i.e., 4000 and 6000) hidden neurons appeared to perform worse than models with 500–2000 hidden neurons. Overall, the results are in line with the well-known observation that the larger the number of training samples, the better the model. The model improvement resulting from increasing the training-set size is roughly constant at 10% when doubling the training set size. Given that the absolute error is the largest of the smallest dataset, we can note that the largest absolute benefit in reducing prediction errors using transfer learning will occur for the smallest training sets.

Figures 1 and 2 show the corresponding numerical results for DNNs with two and three hidden layers, with the complete datasets presented in Tables S2 and S3, respectively, of the Supplementary Information. Similar to the results of one-hidden layer networks in Table 1, these results show that the most important determinant of model quality was the training sample size. Compared to variations in training-set size, the depth and width of the neural networks had a much smaller impact on model performance, especially when there were more than ~ 4000 compounds in the training set.

Table 1 Mean squared error (standard deviation) of test-set compounds for predicting A549 cell inhibition using a single-hidden layer neural network trained as a function of increasing training-set size and with a variable number of neurons in the hidden layer. The data show that models with too few (e.g., 100) or too many (e.g., 6000) hidden neurons do not perform well when trained by small training sets. Doubling the number of compounds in the training set roughly reduced the relative error by 10%. The units of the errors are given in $(\log_{10}(\mathrm{mol/l}))^2$, and the smallest error for each set of training compounds are indicated in boldface font

Number of hidden neurons	Number of training compounds								
	500	1000	2000	3000	4000	10,000	20,000	30,000	40,000
100	0.82 (0.04)	0.74 (0.02)	0.65 (0.02)	0.62 (0.03)	0.59 (0.02)	0.47 (0.02)	0.43 (0.02)	0.40 (0.01)	0.38 (0.01)
500	**0.73 (0.03)**	**0.67 (0.03)**	0.61 (0.03)	0.58 (0.03)	0.55 (0.02)	0.47 (0.01)	0.42 (0.03)	**0.38 (0.01)**	**0.37 (0.01)**
1000	**0.73 (0.04)**	**0.67 (0.02)**	**0.60 (0.02)**	**0.57 (0.03)**	**0.54 (0.03)**	0.46 (0.01)	0.42 (0.02)	**0.38 (0.01)**	0.38 (0.01)
2000	**0.73 (0.04)**	**0.67 (0.02)**	0.61 (0.02)	**0.57 (0.03)**	0.55 (0.03)	**0.45 (0.02)**	**0.41 (0.02)**	0.39 (0.01)	**0.37 (0.01)**
4000	0.74 (0.03)	0.69 (0.02)	0.62 (0.02)	0.58 (0.02)	0.55 (0.02)	0.46 (0.02)	0.42 (0.02)	0.39 (0.01)	**0.37 (0.01)**
6000	0.77 (0.04)	0.70 (0.02)	0.63 (0.02)	0.59 (0.02)	0.56 (0.02)	0.46 (0.01)	**0.41 (0.02)**	0.39 (0.01)	0.38 (0.02)

The smallest error for each set of training compounds are indicated in boldface

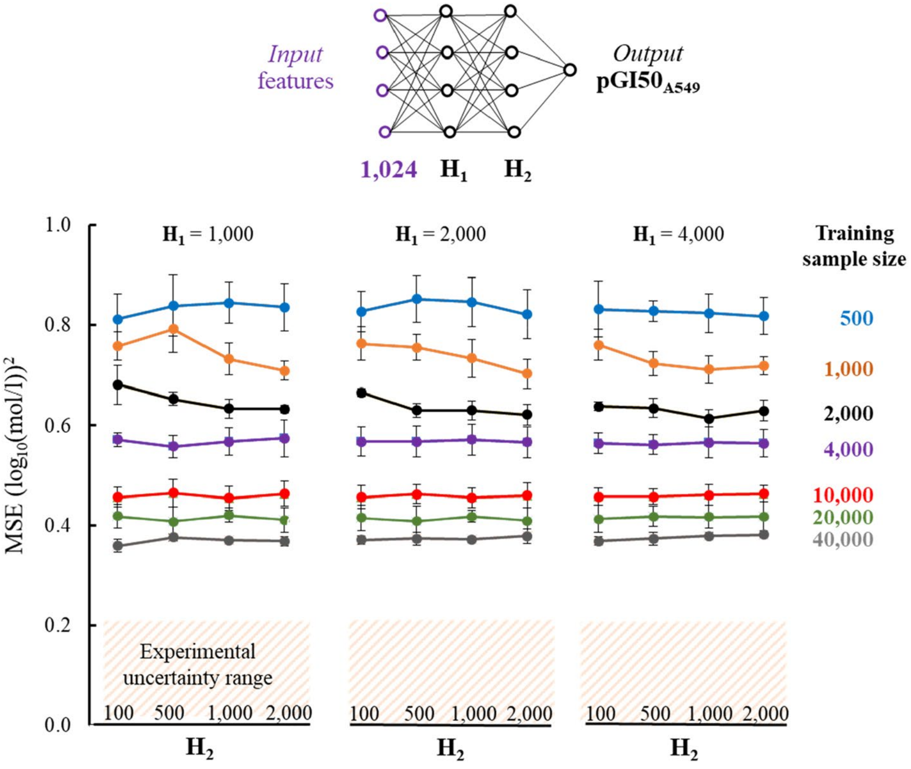


Fig. 1 Mean squared error (MSE) of test-set compounds of two-hidden layer networks trained with an increasing number of compounds. **H$_1$** and **H$_2$** represent the number of neurons in the first and second hidden layers, respectively. For each network architecture and training-set size, we trained 10 models with randomly selected training and validation compounds. Each data point in the figure represents the mean MSE of the test-set compounds calculated over the 10 model predictions, and the vertical bars represent the mean ± 2 standard deviations

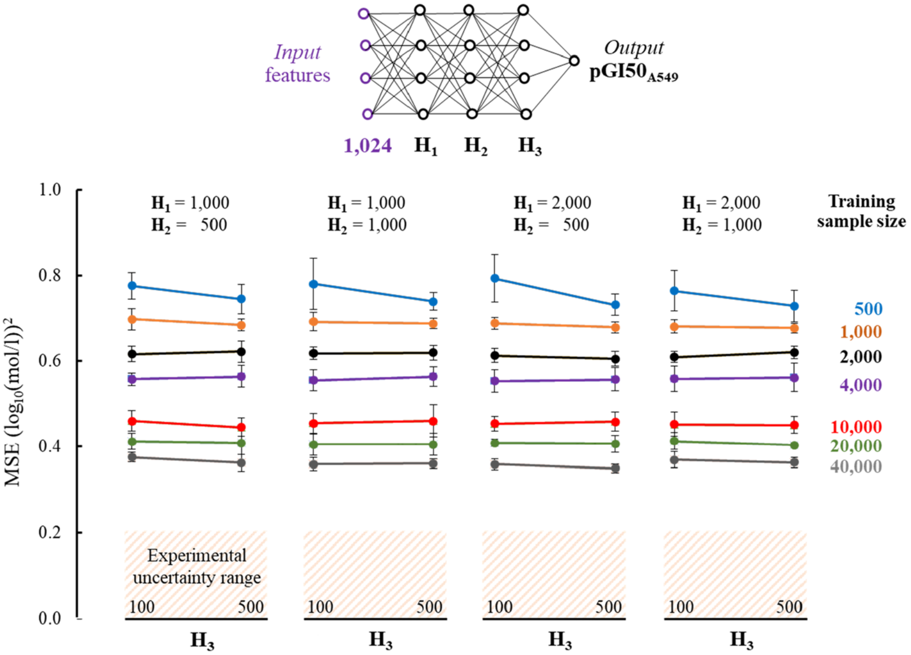


Fig. 2 Mean squared error (MSE) of test-set compounds of three-hidden layer networks trained with an increasing number of compounds. **H$_1$**, **H$_2$**, and **H$_3$** represent the number of neurons in the first, second, and third hidden layers, respectively. For each network architecture and training-set size, we trained 10 models with randomly selected training and validation compounds. Each data point in the figure represents the mean MSE of the test compounds calculated over the 10 model predictions, and the vertical bars represent the mean ± 2 standard deviations

Effect of transferring parameters from a data-rich model to develop a model with limited training data: a proof-of-concept study

To evaluate the effect of transferring parameters from a model trained with a large number of compounds to develop a model with limited experimental data, we initially used A549 data as a data-rich dataset. We first developed a number of different DNN models for predicting pGI$_{50}$ by randomly splitting the dataset into a 90% training set and a 10% validation set to train A549 prediction models with an increasing number of hidden layers. For these models, we used network architectures sized as 1024:1000:1, 1024:1000:1000:1, and 1024:1000:1000:500:1, where the initial nodes of size 1024 correspond to the number of input features, and the final single output node represents the predicted pGI_{50} value. The other integers correspond to the number of hidden neurons in the first, second, and third hidden layers.

Next, we designated the HTB132 (a breast cancer cell line) pGI_{50} data (total number of compounds 5612) to serve as a data-limited dataset. Figure 3 schematically shows the steps executed in evaluating the transfer-learning approach. We randomly selected 10% of the HTB132 data as a test set for evaluating the DNN model performance. From the remaining HTB132 data, we randomly selected 10% as a validation set. We then trained a series of HTB132 models of the same architecture as that of the A549 model using 500, 1000, and 2000 compounds to simulate models trained with small datasets. We also trained a HTB132 model with ~ 80% of the HTB132 dataset (4546 compounds), with the remaining 20% as the validation and test sets, to establish a reference of the best model one could derive from the HTB132 data only (without transfer learning). We used the MSE of the DNN models for the test-set compounds as a performance measure. Finally, we repeated the previous step of training the HTB132 DNN model, but with one to three hidden layers of the A549 models transferred while freezing the values of the weights and biases, and optimizing the rest of the model parameters using the HTB132 training sets. We then calculated the MSE of the test-set compounds using the resulting HTB132 models. Due to the stochastic nature of gradient decent optimization and random assignment of the initial weights and biases, each optimization ended up with a different set of model parameters. We repeated all model training 10 times with randomly selected training and validation compounds to derive statistically reliable results.

Figure 4 shows the results of our evaluation where each data point represents an average of the MSE over the 10 models trained with the same number of randomly selected training samples, where the vertical bar represents ± 2 standard deviations. The three panels show the results as a function of the number of hidden layers in the networks, i.e., N = 1, 2, or 3. The complete datasets are given in Table S4 of the Supplementary Information. Figure 4 (top) shows that, for each network architecture, without transfer learning, model performance depended strongly on the number of compounds in the training set, with the variability decreasing with increasing training-set size, as expected. The range of minimum MSE achievable using the complete HTB132 data could not be reached with the limited-compound training set. However, using the frozen parameters transferred from the A549 model, optimization of the remaining parameters using the same HTB132 training sets resulted in a marked performance improvement, both in terms of considerably smaller average MSEs and their variability. Even with the smallest training set of 500 compounds, transfer learning resulted in considerably better models than training

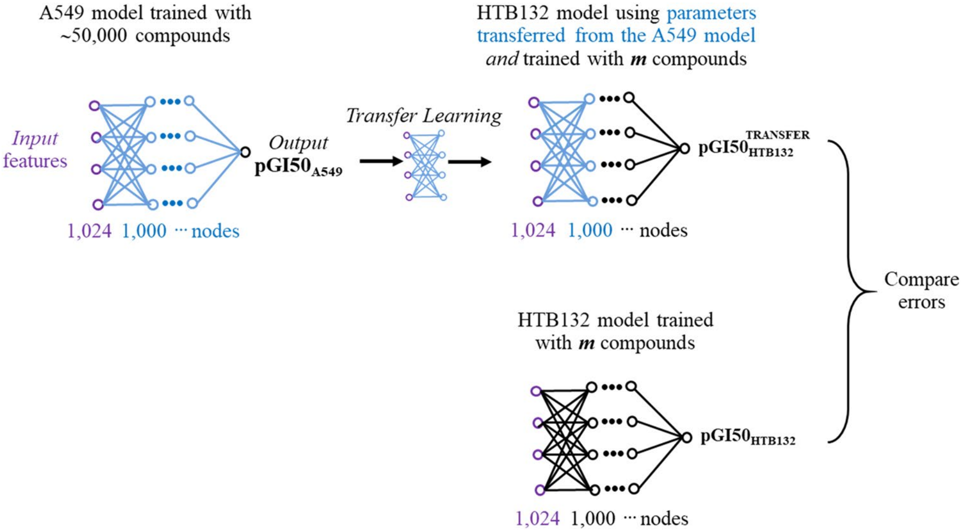


Fig. 3 Scheme of transfer-learning evaluation using datasets of chemical concentrations required to inhibit 50% growth (pGI50) of A549 (human lung cancer) cells and HTB132 (human breast cancer) cells. We first trained a neural network of N hidden layers (with N = 1, 2, or 3) with a large amount of A549 pGI50 data. We transferred the first ***n*** hidden layers, ***n*** = 1, …, N, of the A549 model with frozen weights and biases to construct a HTB132 model of the same architecture. We trained the remaining HTB132 model parameters with pGI50s of ***m*** HTB132 compounds (with ***m*** = 500, 1,000, or 2,000), and calculated the MSE of the HTB132 test set. Finally, we trained a HTB132 model of the same architecture with pGI50s of the same ***m*** compounds, but without transferring any parameters from the A549 model, and calculated the MSE of the test set again. The difference between the MSEs of the two HTB132 models gave an indication of the benefit achieved through of transfer learning

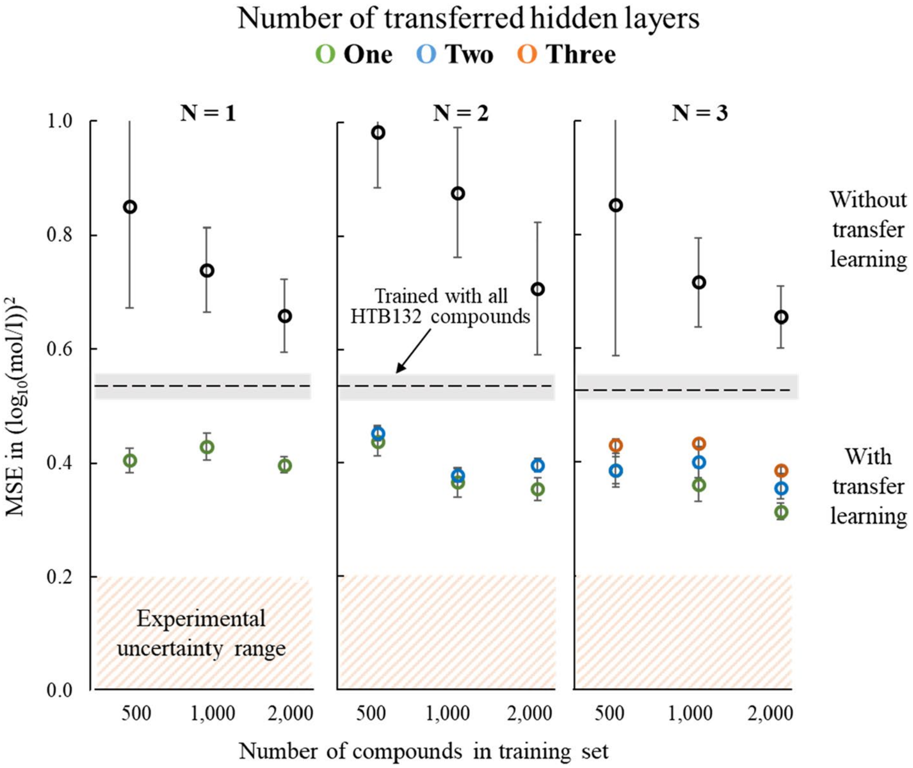


Fig. 4 Mean squared error (MSE) of HTB132 models trained using 500, 1,000, or 2,000 compounds with and without transferring hidden layers from a pre-trained A549 model. Each panel shows the results of a network with a given number of hidden layers (N = 1, 2, or 3). We trained each network architecture 10 times with randomly selected training and validation compounds, resulting in 10 models. Each data point in the figure represents the average MSE over the 10 model predictions, and the vertical bar represents ± 2 standard deviations. The green, blue, and orange circles are the results of transferring one, two, and three hidden layers, respectively. For a 1-hidden layer network, we could transfer learning for at most one hidden layer. For a 3-hidden layer network, we transferred learning for one, two, or all three hidden layers

with all HTB132 compounds without transfer learning. For networks with two or three hidden layers, we transferred parameters for up to three hidden layers, with the results consistently indicating that transfer of the first hidden layer parameters was the most effective. Transferring parameters from additional layers resulted in slightly worse models as judged by the MSE of the test-set compounds. This is most likely due to the presence of more specialized, A549-specific parameters from the A549 growth inhibition DNN model appearing in the second and third hidden layers. Transferring these parameters would not provide any additional benefits to a non A549-specific model, and could instead degrade the prediction performance of the HTB132 model.

Conditions for transfer-learning success and expected benefits

The results of transferring parameters from the A549 model to develop an HTB132 model are promising, yielding results that were better than what could be achieved by using the entire HTB132 dataset itself. The benefits can be partially explained by the high correlation and similarity of the assays themselves, i.e., by measuring chemically induced growth inhibition in cell-line cultures. In fact, the pGI_{50} values of A549 and HTB132 cells were highly correlated with a squared Pearson's correlation coefficient (r^2) of 0.60, as calculated from the 5532 common compounds tested in both growth inhibition assays. As suggested by Xu et al. that assay correlation might be the key to success of multi-task DNN molecular activity models [45], we hypothesized that assay correlation may also be an important contributing factor to the success of transfer learning. Trivially, given an assay correlation of 1.0, transfer learning is by definition the optimal choice of weights. To non-trivially test this hypothesis, we need to assess transfer learning across many pairs of datasets with a broad range of inter-assay r^2 values. Consequently, we selected a number of NCI-60 growth inhibition dataset pairs that included cell lines from different tissue origins and complemented them with additional chemical activity data covering a broad range of inter-assay correlations.

We examined the NCI-60 MALME-3M (a human skin cancer) cell line dataset paired with 28 other cell lines, providing a range of inter-assay pGI_{50} correlations r^2 between 0.45 and 0.87. Similarly, we included the MDA-MB-435 (a human breast cancer) cell line paired with 18 other cell lines, with a range of inter-assay pGI_{50} correlations r^2 between 0.47 and 0.95. Given the nature of the NCI-60 assays and their relatively high correlations ($r^2 > 0.4$), we complemented the NCI-60 dataset pairs with other chemical activity data, such as chemical binding affinity to drug targets, potency to inhibit enzyme functions, as well as physicochemical properties, including lipophilicity and aqueous solubility. Details of these datasets and their pairings are provided in Tables S1 and S5 of the Supplementary Information.

We evaluated transferability of the hidden layers of pre-trained neural networks across the dataset pairs using the 1024:2000:1, 1024:2000:100:1, and 1024:1000:1000:100:1 network architectures, where evaluation procedure followed the steps outlined in Fig. 3.

Thus, for each dataset pair, we designated the larger dataset as the data-rich dataset and the smaller one as the data-limited set. We used a random 90 to 10% split for training and validation of the data-rich models to create the weights and biases of the hidden layers so that they could be transferred for the development of the data-limited models. From each of the data-limited datasets, we first randomly selected 10% of the compounds as a test set. From the remaining compounds, we randomly selected 10% as the validation set. We then randomly selected 500 and 1,000 compounds from the remaining compounds as our data-limited training sets to train neural network models with and without transfer learning. We calculated the MSEs of the test sets using the resulting models and calculated TLE from the MSEs of models trained without and with transfer learning. Figure 5 shows the results for training sets consisting of 500 compounds, and Fig. 6 shows the corresponding data using 1,000-compound training sets. The numerical results are given in Tables S5 and S6 of the Supplementary Information. Figures 5 and 6 are similar, with both showing that when r^2 of a dataset pair was 0.4 or higher, the TLE was larger than zero, and the higher the r^2, the larger the TLE. When r^2 was lower than 0.4, the results were less clear-cut and depended on network architecture. Using the shallow network with a single hidden layer, in a little over 50% of the cases (19 out of 35 with a training set of 500 compounds and 21 out of 35 with a training set of 1000 compounds), transfer learning was able to lower the MSE, as indicated by a TLE > 0. However, using a deeper network with two or three hidden layers, in a majority of the cases, transfer learning resulted in a positive TLE even when r^2 was lower than 0.4.

Figure 7 shows the mean TLE values as a function of r^2 and illustrates that the higher the r^2 between a data-rich and a data-limited dataset, the larger the benefit of transfer learning. The increase in TLE and consequent reduction in prediction error ranged from 10 to 20% for every 0.10 increase in r^2 between the datasets. In the cases where the inter-assay correlations r^2 were lower than 0.4, there was no benefit of using transfer learning for a one-hidden layer network, whereas two- or three-hidden layer network could still benefit.

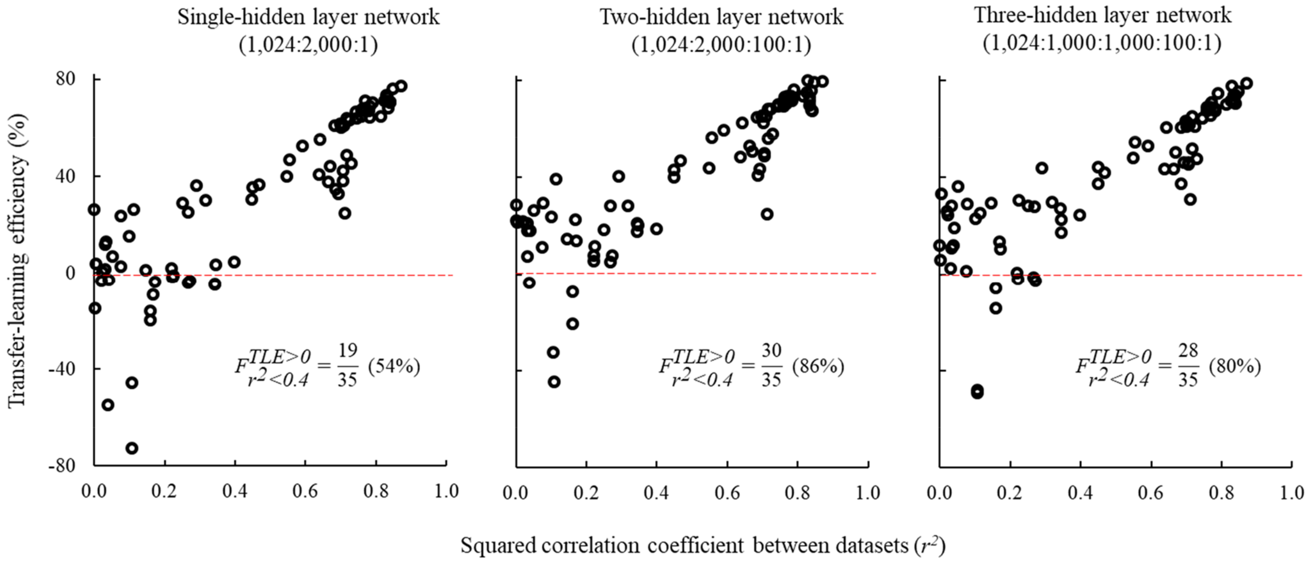


Fig. 5 Transfer-learning efficiency (TLE) vs. squared correlation coefficient (r^2) between datasets. We trained all data-limited models with 500 compounds with or without transferring the first hidden layer from a corresponding data-rich model. The notation $F^{TLE>0}_{r^2<0.4}$ represents the fraction of cases when $r^2 < 0.4$ and TLE was > 0

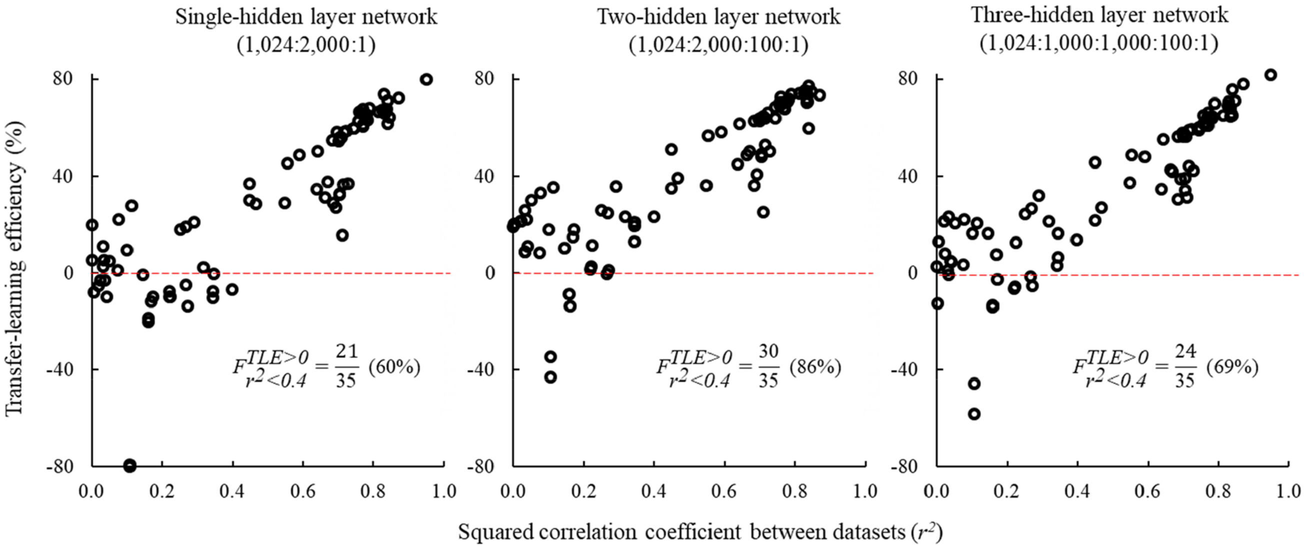
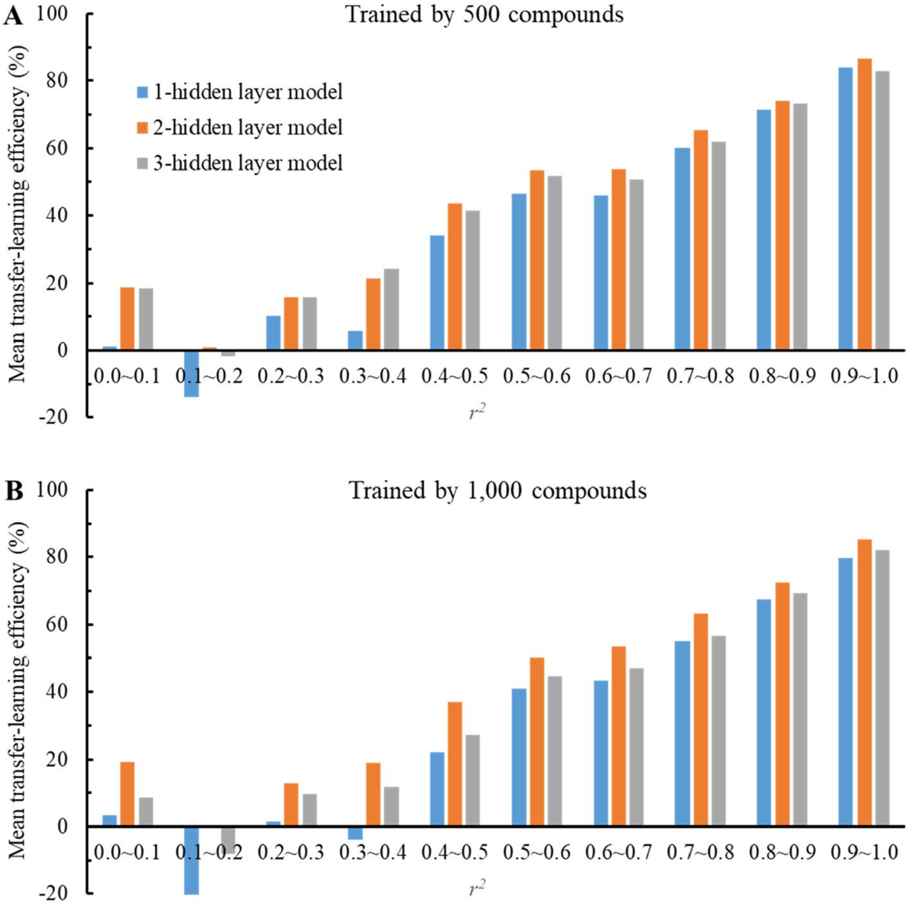


Fig. 6 Transfer-learning efficiency (TLE) vs. squared correlation coefficient (r^2) between datasets. We trained all data-limited models with 1000 compounds with or without transferring the first hidden layer from a corresponding data-rich model. The notation $F^{TLE>0}_{r^2<0.4}$ represents the fraction of cases when $r^2 < 0.4$ and TLE was > 0

Fig. 7 Mean transfer-learning efficiency vs. squared correlation coefficient (r^2) between datasets for neural network models consisting of 1, 2, or 3 hidden layers. We trained the models with either (**A**) 500 or (**B**) 1,000 compounds



Limitations

We introduced the concept of dataset similarity as a metric for deciding when transfer learning could be beneficial to the augment training of DNNs associated with small-size datasets. Currently, we used Pearson's correlation coefficient as a purely numerical evaluation of data similarity, and this may not capture all considerations for evaluating transfer learning. Furthermore, we do not know the correlation metric a priori, as it has to be estimated from the datasets themselves based on a potentially limited number of compounds tested in both datasets. Although this can be a practical limitation when confronted with narrow chemical diversity among the data, for chemical property applications based on minima; datasets of ~ 10^2 compound, the transfer learning approach described here might be the only practical way forward to implement a data-driven prediction model.

Summary

The goal of this study was to evaluate if and under what conditions the dense layers of a pre-trained DNN can be transferred and used for the development of another DNN associated with limited training data. Our results derived from molecular activity data indicated that, unlike the convolutional layers of a DNN for image recognition where all layers are considered transferrable, the bulk of error reduction in developing a network using a small dataset was associated with transfers from the first dense layer. Transfer of other dense layers did not result in additional benefit, suggesting that deeper, dense layers conveyed more specialized and assay-specific information. In addition, the benefits of transferring the first dense layer were related to the extent of inter-assay dataset correlation. The larger the correlation, the higher the transfer-learning efficiency. Interestingly, even when there was no apparent correlation, or when there was very low correlation between two datasets, transfer learning of DNNs with two or three hidden layers was still beneficial, albeit with a lower reduction of model error.

Note that in this study we used training sets of 500 and 1,000 compounds to simulate small training sets. Results of our evaluation of the impact of training-set size indicated that the larger the training set, the better the resulting model regardless of network architecture. Thus, we can reasonably expect that transfer-learning efficiency will decrease with increasing training-set size. On the other hand, with an increasing amount of training data, there is a decreasing need for transfer learning. Therefore, the transfer-learning strategy evaluated in this study may be useful for partially overcoming the challenge of deep learning with "small" datasets, when only a limited number of compounds have been tested for their potency at a drug target, or for in vivo studies where only a limited number of compounds can be assayed.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10822-022-00486-x.

Authors' contributions All authors contributed to the study conception and design. Material preparation, data collection, computation, and analysis were performed by Ruifeng Liu and Srinivas Laxminarayan. The first draft of the manuscript was written by Anders Wallqvist and Ruifeng Liu. All authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research was funded by the U.S. Army Medical Research and Development Command under Contract No. W81XWH20C0031 and by Defense Threat Reduction Agency Grant CBCall14-CBS-05-2-0007.

Availability of data and materials All the data used in this study were downloaded from public repositories as described in the Molecular Activity Datasets Section. The web links to specific datasets are given in Table S1 in the Supplemental Materials. We generated the input features using Pipeline Pilot 18.1.100.11 (Dassault Systèmes, Vélizy-Villacoublay, France, an evaluation license is available from https://www.3ds.com/how-to-buy/contact-sales). For anyone interested in repeating the computations but having no access to Pipeline Pilot, we provided the input data generated from Pipeline Pilot in Supplementary Information. We performed all DNN studies using the open source Keras API in Tensorflow 2.1.0 available from https://www.tensorflow.org/. Our python code for transfer learning using a two-hidden layer neural network is provided in the Supplementary Information.

Declarations

Competing interests The authors declare no competing interests.

Consent for publication All authors have given consent for publication of the article. The opinions and assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the U.S. Army, the U.S. Department of Defense, or The Henry M. Jackson Foundation for the Advancement of Military Medicine, Inc. This paper has been approved for public release with unlimited distribution.

References

1. Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, Ferrero E, Agapow PM, Zietz M, Hoffman MM, Xie W, Rosen GL, Lengerich BJ, Israeli J, Lanchantin J, Woloszynek S, Carpenter AE, Shrikumar A, Xu J, Cofer EM, Lavender CA, Turaga SC, Alexandari AM, Lu Z, Harris DJ, De Caprio D, Qi Y, Kundaje A, Peng Y, Wiley LK, Segler MHS, Boca SM, Swamidass SJ, Huang A, Gitter A, Greene CS (2018) Opportunities and obstacles for deep learning in biology and medicine. J R Soc Interface 15:20170387
2. Loiodice S, Nogueira da Costa A, Atienzar F (2019) Current trends in in silico, in vitro toxicology, and safety biomarkers in early drug development. Drug Chem Toxicol 42:113–121
3. Muster W, Breidenbach A, Fischer H, Kirchner S, Muller L, Pahler A (2008) Computational toxicology in drug development. Drug Discov Today 13:303–310
4. Valerio LG Jr (2009) In silico toxicology for the pharmaceutical sciences. Toxicol Appl Pharmacol 241:356–370

5. Keyvanpour MR, Shirzad MB (2021) An analysis of QSAR research based on machine learning concepts. Curr Drug Discov Technol 18:17–30
6. Piir G, Kahn I, Garcia-Sosa AT, Sild S, Ahte P, Maran U (2018) Best practices for QSAR model reporting: physical and chemical properties, ecotoxicity, environmental fate, human health, and toxicokinetics endpoints. Environ Health Perspect 126:126001. https://doi.org/10.1289/EHP3264
7. Tropsha A, Golbraikh A (2007) Predictive QSAR modeling workflow, model applicability domains, and virtual screening. Curr Pharm Des 13:3494–3504
8. Neves BJ, Braga RC, Melo-Filho CC, Moreira-Filho JT, Muratov EN, Andrade CH (2018) QSAR-based virtual screening: advances and applications in drug discovery. Front Pharmacol 9:1275. https://doi.org/10.3389/fphar.2018.01275
9. Mao J, Akhtar J, Zhang X, Sun L, Guan S, Li X, Chen G, Liu J, Jeon HN, Kim MS, No KT, Wang G (2021) Comprehensive strategies of machine-learning-based quantitative structure-activity relationship models. iScience 24:103052. https://doi.org/10.1016/j.isci.2021.103052
10. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inform 29:476–488
11. Shaikhina T, Khovanova NA (2017) Handling limited datasets with neural networks in medical applications: a small-data approach. Artif Intell Med 75:51–63
12. Sosnin S, Vashurina M, Withnall M, Karpov P, Fedorov M, Tetko IV (2019) A survey of multi-task learning methods in chemoinformatics. Mol Inform 38:e1800108. https://doi.org/10.1002/minf.201800108
13. Deng J, Dong W, Socher R, Li L, Li K, Li F (2009) ImageNet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
14. Emmert-Streib F, Yang Z, Feng H, Tripathi S, Dehmer M (2020) An introductory review of deep learning for prediction models with big data. Front Artif Intell 3:4. https://doi.org/10.3389/frai.2020.00004
15. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444
16. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q (2021) A comprehensive survey on transfer learning. Proc IEEE 109:43–76
17. Zhuang D, Ibrahim AK (2021) Deep learning for drug discovery: a study of identifying high efficacy drug compounds using a cascade transfer learning approach. Appl Sci 11:7772. https://doi.org/10.3390/app11177772
18. Li Y, Xu Y, Yu Y (2021) CRNNTL: convolutional recurrent neural network and transfer learning for QSAR modeling in organic drug and material discovery. Molecules 26:7257. https://doi.org/10.3390/molecules26237257
19. Yamda H, Liu C, Wu S, Koyama Y, Ju S, Shiomi J, Morikawa J, Yoshida R (2019) Predicting materials properties with little data using shotgun transfer learning. ACS Cent Sci 5:1717–1730
20. Cai C, Wang S, Xu Y, Zhang W, Tang K, Quyang Q, Lai L, Pei J (2020) Transfer learning for drug discovey. J Med Chem 63:8683–8694
21. Hu S, Chen P, Gu P, Wang B (2020) A deep learning-based chemical system for QSAR prediction. IEEE J Biomed Health Inform 24:3020–3028
22. Fernandez-Torras A, Comajuncosa-Creus A, Duran-Frigola M, Aloy P (2022) Connecting chemistry and biology through molecular descriptors. Curr Opin Chem Biol 66:102090. https://doi.org/10.1016/j.cbpa.2021.09.001
23. Chuang KV, Gunsalus LM, Keiser MJ (2020) Learning molecular representations for medicinal chemistry. J Med Chem 63:8705–8722
24. Xue L, Bajorath J (2000) Molecular descriptors in chemoinformatics, computational combinatorial chemistry, and virtual screening. Comb Chem High Throughput Screen 3:363–372
25. Sahoo S, Adhikari C, Kuanar M, Mishra BK (2016) A short review of the generation of molecular descriptors and their applications in quantitative structure property/activity relationships. Curr Comput Aided Drug Des 12:181–205
26. Rogers D, Hahn M (2010) Extended-connectivity fingerprints. J Chem Inf Model 50:742–754
27. Broccatelli F, Trager R, Reutlinger M, Karypis G, Li M (2022) Benchmarking accuracy and generalizability of four graph neural networks using large in vitro ADME datasets from different chemical spaces. Mol Inform. https://doi.org/10.1002/minf.202100321
28. Carracedo-Reboredo P, Linares-Blanco J, Rodriguez-Fernandez N, Cedron F, Novoa FJ, Carballal A, Maojo V, Pazos A, Fernandez-Lozano C (2021) A review on machine learning approaches and trends in drug discovery. Comput Struct Biotechnol J 19:4538–4558
29. Deng D, Chen X, Zhang R, Lei Z, Wang X, Zhou F (2021) XGraphBoost: extracting graph neural network-based features for a better prediction of molecular properties. J Chem Inf Model 61:2697–2705
30. Jiang D, Wu Z, Hsieh CY, Chen G, Liao B, Wang Z, Shen C, Cao D, Wu J, Hou T (2021) Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. J Cheminform 13:12. https://doi.org/10.1186/s13321-020-00479-8
31. Yang K, Swanson K, Jin W, Coley C, Eiden P, Gao H, Guzman-Perez A, Hopper T, Kelley B, Mathea M, Palmer A, Settels V, Jaakkola T, Jensen K, Barzilay R (2019) Analyzing learned molecular representations for property prediction. J Chem Inf Model 59:3370–3388
32. Wieder O, Kohlbacher S, Kuenemann M, Garon A, Ducrot P, Seidel T, Langer T (2020) A compact review of molecular property prediction with graph neural networks. Drug Discov Today Technol 37:1–12
33. Sun M, Zhao S, Gilvary C, Elemento O, Zhou J, Wang F (2020) Graph convolutional networks for computational drug development and discovery. Brief Bioinform 21:919–935
34. Shoemaker RH (2006) The NCI60 human tumour cell line anticancer drug screen. Nat Rev Cancer 6:813–823
35. Close DA, Wang AX, Kochanek SJ, Shun T, Eiseman JL, Johnston PA (2019) Implementation of the NCI-60 human tumor cell line panel to screen 2260 cancer drug combinations to generate >3 million data points used to populate a large matrix of anti-neoplastic agent combinations (ALMANAC) database. SLAS Discov 24:242–263
36. Liu T, Lin Y, Wen X, Jorissen RN, Gilson MK (2007) BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. Nucleic Acids Res 35:D198-201
37. Wang Y, Bryant SH, Cheng T, Wang J, Gindulyte A, Shoemaker BA, Thiessen PA, He S, Zhang J (2017) PubChem BioAssay: 2017 update. Nucleic Acids Res 45:D955–D963
38. Gadaleta D, Vukovic K, Toma C, Lavado GJ, Karmaus AL, Mansouri K, Kleinstreuer NC, Benfenati E, Roncaglioni A (2019) SAR and QSAR modeling of a large collection of LD50 rat acute oral toxicity data. J Cheminform 11:58. https://doi.org/10.1186/s13321-019-0383-2
39. Sorkun MC, Khetan A, Er S (2019) AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. Sci Data 6:143. https://doi.org/10.7910/DVN/OVHAW8
40. Bergstra J, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. In Advances in neural information processing systems 2546–2554.

41. Ma J, Sheridan RP, Liaw A, Dahl GE, Svetnik V (2015) Deep neural nets as a method for quantitative structure-activity relationships. J Chem Inf Model 55:263–274
42. Ramsundar B, Liu B, Wu Z, Verras A, Tudor M, Sheridan RP, Pande V (2017) Is multitask deep learning practical for pharma? J Chem Inf Model 57:2068–2076
43. Kingma DP, Ba JL (2015) Adam: A Method for Stochastics Optimization. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA. https://arxiv.org/pdf/1412.6980.pdf.
44. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th international conference on artificial intelligence and statistics, Chia Laguna Resort, Sardinia, Italy 2010. Volume 9 of JMLR: W&CP 9. http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf
45. Xu Y, Ma J, Liaw A, Sheridan RP, Svetnik V (2017) Demystifying multitask deep neural networks for quantitative structure-activity relationships. J Chem Inf Model 57:2490–2504