Massively Multitask Networks for Drug Discovery: Review

Team Live and Learn

Background

With the advent of machine learning, there have been numerous attempts to use these techniques to ease and increase efficiency of drug discovery process. However, all the existing methods face various challenges in this space. Given a target, finding a compound active on it is hard and hit rate on experimental screen is often only 1-2%. This introduces class imbalance in datasets which calls for extra care in designing an effective learning model. In this paper, Ramsundar et al. lay out a learning framework for drug discovery using massively multitask neural networks. Deep learning allows for the synthesis of large amounts of data from different sources, and in the context of drug discovery, multitask networks, by sharing information across different experiments, can potentially overcome the disadvantage of limited data from any single experiment under consideration.

In this paper, the authors show that multitask networks achieve significant improvements over existing baseline machine learning methods for drug discovery. Moreover, their empirical studies suggest that increasing the total amount of tasks and data both improve the predictive power of multitask network. While previous works (e.g. Dahl 2012, Unterthiner et al.) have explored the applications of multitask neural networks to drug discovery, this is the first comprehensive exploration of the effects of massively multitask networks and the independent effects of larger datasets. This paper shows that the effects of multitasking scale even to over 200 targets, and, more importantly, that the effects of multitasking and larger datasets never “plateau”; they continue to climb as more tasks and data are added. This is an invaluable finding in such an important field.

Methods

The authors train the models using 259 datasets from multiple publicly available datasets of drug virtual screening. They do not perform any preprocessing. To featurize each molecule, the authors use extended connectivity fingerprints generated by RDKit. Specifically, the molecules are converted to a set of fragments, where each of them is centered at a non-hydrogen atom. Then each fragment is assigned a unique identifier, and the collection of identifiers for a molecule is hashed into a fixed-length bit vector to construct the molecular fingerprint.

Each task in multitasking corresponds to the classifier associated with a particular dataset. To make the neural network capable of multitasking, the authors use multiple softmax classifiers at the final layers, one for each task, whereas all weights in other layers are shared across the tasks. The neural network architecture is shown in Figure 1.

The key metrics they use when considering multiple datasets are the mean and median of the k-fold-average area under the ROC curve (AUC). Note that due to the lack of standard metrics, it is

difficult to compare the results with previous work in the literature. There is an alternative metric called "enrichment", which the authors also include in the appendix for completeness.
Overall, while this section is well-explained, we note that the standard deviation for the % of active compounds in each dataset is high compared to the mean values. This might invalidate the claim about the correlation between the correlation between multitask improvement and %active of datasets. This could imply a multiple hypothesis testing issue.

Experimental Section

Experimental Exploration of Massively Multitask Networks

The authors tested various models with various hyperparameters. They found that their best multitask architecture significantly outperformed all other single-task models. One major obstacle the authors encountered early on was the small number of positive examples. This led to high overfitting, which was alleviated by the pyramidal net architecture the authors devised. The first hidden layer is extremely wide (2000 nodes), which allows for complex features to be learned, while the second hidden layer is extremely narrow (100 nodes), which provides dimensionality reduction. The authors also used a dropout rate of .25, which also improved performance. When the authors ran the pyramidal architecture on single-task models, they found that performance exceed that of ordinary neural nets, but was still inferior to performance on multitask models. This demonstrates that the pyramidal scheme is a very promising architecture that can be beneficial for small datasets.

More tasks vs more data

In order to understand how the number of tasks and the training data size affect performance, the authors train and compare different models by varying the number of tasks as well as the size of the training data. This experiment is performed very rigorously; the authors train models with a cumulatively increasing number of tasks and a cumulatively increasing training data set size. Since these factors are changed independently of each other, it is possible to compare their effects. In doing so, the authors find that both factors have an impact on performance. Specifically, if the number of tasks is held constant, increasing the size of the training dataset helps, and if the dataset size is held constant, increasing the number of tasks helps. Overall, the authors find that the performance of multitask networks never plateaus (on average) as more tasks are added; this is an extremely promising finding and a strong case for the future use of multitask networks in this field. However, it seems like performance does in fact plateau for some datasets; more interestingly, there are visible dips in performance at a certain # of tasks for some datasets. While the authors attempt to explain this in part when they explore the reasons for the success of these networks, it would be interesting to do a deep dive into the specific combinations of tasks that caused performance to drop for some datasets and not others.

Generalizability

The authors also explore the generalizability and reusability of massively multitask models through transfer learning. Specifically, they hold out ten datasets from section 4.2, and use the learned weights from existing models to initialize (and then finetune) single task networks on those datasets. They show that doing so adversely affects many of the models; many of them perform significantly below the

baseline, and thus they conclude that the effect of transfer learning is negative. They also point out that larger multitask networks exhibit more positive effects of transfer learning.

Reasons for success of multitask training

In this section, the authors attempt to explore the causes behind the success of multitask learning; specifically, what are the properties that cause multitask learning to be more successful for some datasets than for others?

One of the first causes explored is the presence of shared compounds. Since active compounds are so underrepresented in the datasets, the authors hypothesize that they must contain more information than inactive compounds, and thus the number of shared active compounds can be thought of as a measure of dataset similarity. Their overarching goal is to show that dataset similarity leads to more successful multitasking. To demonstrate this, they show how much improvement multitasking yields, as a function of the mean "active occurrence rate" (the number of datasets among which an active compound is shared) for each dataset (excluding DUD-E). This plot shows a mild correlation (0.33) between AOR and multitask improvement; however, the authors point out that the inclusion of DUD-E causes the correlation to drop to 0.22. This seems worthy of investigation, especially since the DUD-E dataset (as explained by the authors) consistently has the highest AUC and is essential to improving performance on the other datasets.

The second question they attempt to answer is whether certain biological targets benefit more from multitasking or not. They show very straightforwardly that the answer is no; multitask learning benefits all target classes to the same extent.

Finally, the authors explore the effect of duplicate targets on performance. In particular, they are interested in whether targets that are duplicated across tasks benefit more from multitasking or not. The analysis that they perform here is very thorough, and shows that the statistical effects of duplicated targets are most likely very low, and thus can be ignored in this paper.

Discussion and Conclusion

This paper investigated multitask networks for virtual screening using models trained on publicly available data. The networks achieved significant improvement over basic non-neural-network machine learning methods. Most significantly, multitask performance improved by adding more tasks and more data. Multitask learning allows for limited learning transfer, but it requires a large amount of data to show the effect. Multitask improvement also works best on highly similar datasets (i.e. in the presence of shared active compounds). This work is novel. Other applications of deep learning use different datasets that are not directly comparable. We cannot compare this method to existing ones. Other interesting directions that could be explored are the use of unsupervised learning to explore more of the chemical space, as well as other featurization techniques.

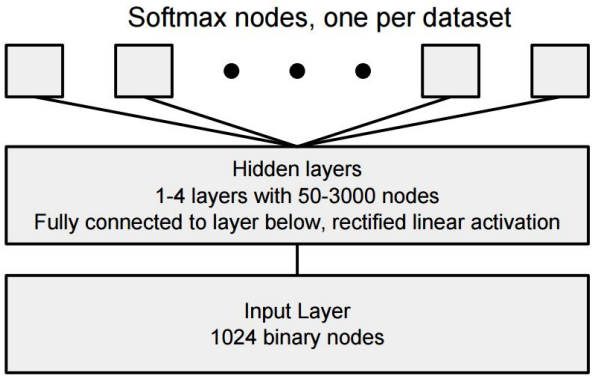

Figure 1. Multitask neural network.

References

Dahl, George. Deep Learning How I Did It: Merck 1st place interview. No Free Hunch, November 1, 2012.

Unterthiner, Thomas, Mayr, Andreas, u ñter Klambauer, G, Steijaert, Marvin, Wenger, Jo ŕg, Ceulemans, Hugo, and Hochreiter, Sepp. Deep learning as an opportunity in virtual screening.