

DeepMTP: a Python-based deep learning framework for multi-target prediction^{*}

Dimitrios Iliadis¹[0000-0002-3676-5940], Bernard De Baets¹[0000-0002-3876-620X],
and Willem Waegeman¹[0000--0002-5950-3003]

KERMIT, Department of Data Analysis and Mathematical Modelling, Ghent
University, Coupure links 653, B-9000 Ghent, Belgium
{dimitrios.iliadis,bernard.debaets,willem.waegeman}@ugent.be

Abstract. DeepMTP is a python framework designed to be compatible with the majority of machine learning sub-areas that fall under the umbrella of multi-target prediction (MTP). These include problem settings like multi-label classification, multivariate regression, multi-task learning, matrix completion, dyadic prediction, and zero-shot learning. Even though there exist major similarities between these problem settings, they've ended up developing in separate areas. Instead of using separate methodologies for the different problem settings, the proposed framework employs a generic and flexible two-branch neural network architecture that has been proven to be effective across the majority of MTP problem settings. To our knowledge, this is the first attempt at providing a framework that is compatible with more than two MTP problem settings in a simplified package that supports multiple input formats, performance metrics, and logging options. The source code of the framework is available at <https://github.com/diliadis/DeepMTP> and an extension with a user interface that requires no coding experience is available at https://github.com/diliadis/DeepMTP_gui.

Keywords: Multi-target prediction · Multi-label classification · Multivariate regression.

1 Introduction

Over the last two decades, the machine learning community has delivered numerous libraries with implementations of popular models. Older libraries like WEKA [3] and scikit-learn [7] implemented more traditional machine learning methods focused on standard classification, regression and clustering tasks. More recently, python-based libraries like Pytorch [6] and Tensorflow [1] focused on neural networks, resulting in a substantial gain of popularity among machine learning practitioners.

This packages have lowered the barrier to entry for many ML practitioners, providing them with the ability to avoid the technical and time-consuming implementation details and instead focus on the higher-level aspects of modeling

^{*} Supported by organization x.

and evaluation. The aforementioned libraries provide this higher lever of abstraction for regular classification, regression, and clustering tasks but typically require lower-level control if a user intends to implement more advanced tasks. Multi-target prediction can be seen as an umbrella term that groups together supervised learning tasks that share one basic property, the simultaneous prediction of multiple target variables of diverse type. Even though this area as a whole is less well known, specific problem settings like multi-label classification and multivariate regression are prevalent in modern tasks like that of image tagging in the computer vision field [2, 10].

Furthermore, in terms of published software, the options are limited. One of the most popular choices is the MULAN library [9] which contains an extensive selection of models but is limited both in terms of the MTP problem settings it supports, as well as the now less popular Java programming language it uses. Another popular library that offers compatibility with the scikit/scipy ecosystem in a Python package is the scikit-multilearn library [8]. Due to the limited space in this work, we will present a more extensive review of MTP software packages in a future publication.

The baseline architecture of the DeepMTP framework was popularized by the neural collaborative filtering (NCF) framework [4] in the field of recommender systems. In our work [5] we successfully showed that the same general architecture can be adapted with minimal modifications and achieve competitive performance in six MTP problem settings across 21 different datasets. The adaptability of the framework can be summarized in three main aspects. Depending on the availability of side-information (features) for the instances and targets, the framework can either use the provided information or create one-hot encoded vectors in case it's missing. Furthermore, the framework can adapt to different input types (tabular, images, time-series) using different sub-architectures inside the two branches (fully-connected layers, convolutional architecture, recurrent neural networks). Finally, the pairwise manner in which the framework processes samples enables it to support all four possible MTP validation settings.

Our python package implements the basic architecture and offers additional functionality that automates multiple steps in the standard machine learning pipeline. In the data preparation step, the package automatically determines characteristics like data availability, validation setting, type of prediction task (classification, regression), all of which help to determine the specific MTP problem setting and the subsequent modifications of the two-branch neural network. In terms of the sub-architectures in the two branches, users have the option to use the ones already implemented in the package, or even provide their own. Finally, for the hyperparameter optimization step, we provide methods like the standard random search and grid search, as well as the more advanced hyperband approach. A detailed benchmarking of different hyperparameter optimization methods on multiple MTP problem settings using DeepMTP will be published in the near future. To conclude, DeepMTP is a python framework that offers a flexible neural network architecture capable of working with multiple MTP problem settings.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: {TensorFlow}: a system for {Large-Scale} machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). pp. 265–283 (2016)
2. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5177–5186 (2019)
3. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter **11**(1), 10–18 (2009)
4. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. pp. 173–182 (2017)
5. Iliadis, D., De Baets, B., Waegeman, W.: Multi-target prediction for dummies using two-branch neural networks. Machine Learning **111**(2), 651–684 (2022)
6. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
8. Szymanski, P., Kajdanowicz, T.: Scikit-multilearn: a scikit-based python environment for performing multi-label classification. The journal of machine learning research **20**(1), 209–230 (2019)
9. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. The Journal of Machine Learning Research **12**, 2411–2414 (2011)
10. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: Cnn-rnn: A unified framework for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2285–2294 (2016)