2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Transfer Meta Learning

Nico Zengeler
Hochschule Ruhr West
Computer Science Institute
Bottrop, Germany
Email: nico.zengeler@hs-ruhrwest.de

Tobias Glasmachers
Ruhr Universität Bochum
Institut für Neuroinformatik
Bochum, Germany
Email: tobias.glasmachers@ini.rub.de

Uwe Handmann
Hochschule Ruhr West
Computer Science Institute
Bottrop, Germany
Email: uwe.handmann@hs-ruhrwest.de

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956622

***Abstract*—Transfer Learning methods aim to reuse previously acquired knowledge about a source task to facilitate learning of a target task. In this paper, we present a Meta Learning approach to find optimal hyperparameters for Transfer Learning processes given previously known metadata about the source task, the target task, and the pre-trained model. We collected metadata and model parameters from more than 15,000 Transfer Learning processes in a dataset, which we use to learn metamodels that predict a Transfer Learning process result in terms of accuracy on the validation sets, given prior information such as the number of epochs, learning rates, optimizers, etc. Using feedforward multilayer perceptrons (MLP), we show that and how our approach finds efficient hyperparameters for Transfer Learning for image classification.**

I. Introduction

With the increasing popularity of Deep Learning, new applications arise as new data becomes available. However, optimizing deep neural networks can take a lot of computational time and thus energy when learning from scratch. With Transfer Learning methods, we seek to reduce this cost by reusing previously acquired knowledge to facilitate the learning problem and thus speed up the underlying computational process. However, the problem of model selection and hyperparameter selection for the Transfer Learning process remains and still requires expert knowledge to find useful learning processes. We address the problem of model selection and hyperparameter selection with Meta Learning based on metadata from Transfer Learning processes. Our approach uses multilayer perceptrons (MLP) to approximate quantities we know after a Transfer Learning process, e.g., the accuracy on the validation set, given relevant information we know in advance. Since this means that we can use fewer computational resources to systematically achieve good Transfer Learning results, it also means that we need to use less electrical energy to perform our Deep Learning computations, ultimately reducing carbon dioxide emissions for computing.

With the methodology summarized in Figure 1, we seek to answer the research question of whether we can use a systematic Meta Learning procedure to obtain suggestions for Transfer Learning processes that lead to better results than simply using the best performing setting so far or a prediction by linear regression. In order to answer the research question, we first created a meta dataset of $15,972$ Transfer Learning processes and then examined the learnability of the recorded relationships by conducting a total of $10,402$ Meta Learning experiments, which we evaluated and compared to the baseline. We conclude that our Transfer Meta Learning approach constitutes a promising contribution. To the best of our knowledge, we designed a novel approach, which relies on the fundamental idea to collect useful metadata from Transfer Learning processes in the first place and then use them to estimate better hyperparameters than the current state of the art.

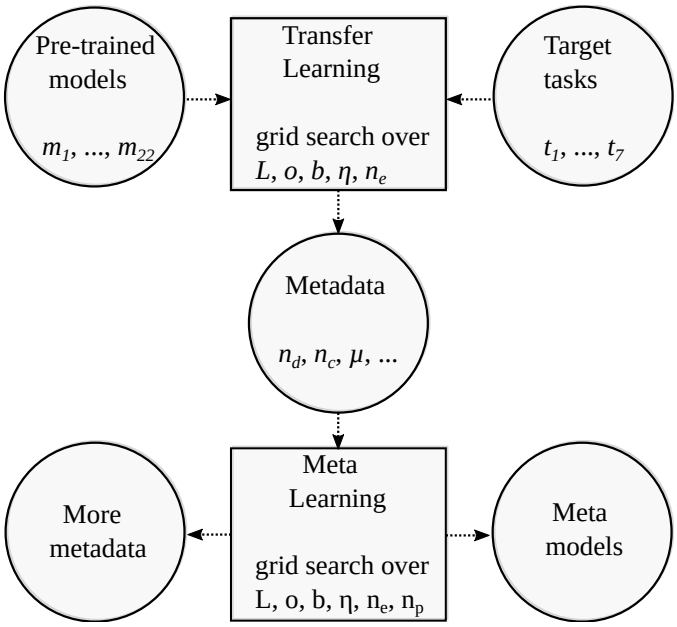

Fig. 1: The methodical sequence of our studies in the form of a flowchart.

Contribution: Motivated by the open questions from [1] concerning optimal Transfer Learning parametrization, we present a systematic approach to optimize Transfer Learning processes. We work on a tool that can answer the question of which model leads to high accuracies under which Transfer Learning parameters with knowledge of metadata alone, so without knowing any actual samples from the target task's dataset. This tool should reduce the need for expert knowledge for the application of Transfer Learning and contributes pro-

posals to finding better learning processes than straightforward methods. In this paper, we limit our investigation to the image classification problem and the Transfer Learning method of fine-tuning only, but we can well imagine applications of the underlying approach to other problems of Machine Learning. Our contribution differs from related research, which we discuss in Section II, first of all in the research question but also in the concrete implementation. In section III, we detail the methodology we use to build our dataset that we conduct Transfer and Meta Learning experiments with. After discussing the results of our experiments in relation to our objective in section IV, we limitations and future work in section V.

II. Related work

Transfer Learning: In Transfer Learning methods, Machine Learning models use knowledge from a source domain to enhance, accelerate, or stabilize learning in a target domain [1], [2], [3], [4], [5], [6], [7]. The idea of Transfer Learning also relates to the concepts of Continual Learning [8], Multitask Learning [9], Semi-Supervised Learning [10], [11] and Knowledge Distillation [12], [13]. Depending on the application, different Transfer Learning techniques suit use cases, for instance Few Shot Learning for classification tasks [14] or Reinforcement Learning tasks [15], robotics [16], [17], person identification [18], [19], finding visual analogies [20] as well as Adversarial Reprogramming [21] and Natural Language Processing [22]. Catastrophic forgetting, a well-studied problem of Transfer Learning, makes it difficult to learn knowledge of the target domain while maintaining knowledge of the source domain [23], [24], [25]. To name some useful applications of Transfer Learning, for example, it helps to reduce the overexploitation of natural resources by supporting circular economy [26], [27], it helps in medical imaging and diagnostics [28] and in work safety [29], [30].

Meta Learning: For our purposes, the concepts of Meta Learning deal with how a machine learns Machine Learning, i.e., what methods with what hyperparameters lead to good results [31], [32]. The general concepts also apply to other fields; for example in Natural Science, applications that learn to transfer simulation settings from metadata save computational time and energy [33], [34], [35].

Transfer Meta Learning: The current state of the art already knows several approaches that combine the concepts of Transfer Learning and Meta Learning. In the context of Deep Learning, learning generalized meta representations can help to find highly transferable parameter vectors for Zero-Shot and Few-Shot approaches [36], [37], [38], [39], such that we can also consider meta representation learning itself as a method of Transfer Learning. The search for a good Representation Learning algorithm led to a benchmark for Transfer Learning, the Visual Task Adaptation Benchmark (VTAB) [40]. In an application concerning adaptive beamforming optimization for a signal processing problem, the combination of Transfer and Meta Learning helps to solve the problem of performance deterioration when the testing environment changes [41]. Another approach uses the idea of Meta Learning kernels to chain transformations to help a Transfer Learning to quickly learn new target tasks [42]. A demonstration of a Transfer Meta Learning ensemble with fast sigmoidal regression models that outperform state-of-the-art approaches on a certain data set uses evolved hierarchical ensembles as building blocks for Meta Learning [43]. Information theory considerations explore the limitations of Transfer Meta Learning in the form of a meta learner knowing data from source tasks while evaluating the performance on new target tasks and conclude that upper bounds for of Empirical Meta-Risk Minimization lie in the average generalization gap, the high probability Bayesian bounds and the high probability single draw bounds [44].

III. Method

Our method, which we refer to as Transfer Meta Learning, starts with a set of pre-trained models that we wish to fine-tune to a set of tasks and ends with metamodels that we use to map the resulting a priori and the a posteriori metadata. With it, we aim to systematically achieve Transfer Learning proposals better than simply using the best-known settings, as motivated in [1], or the ones we would derive by using linear regression. Therefore, we perform a grid search of Transfer Learning processes and record the individual outcomes and relevant metadata. Figure 1 provides a brief overview of the process and interrelationships of our studies. Based on these metadata, we perform Meta Learning to obtain metamodels that predict the a posteriori information about the Transfer Learning processes, such as the accuracies on the respective training, validation and test sets of the target task, given the a priori knowledge, such as the identity of the model, the number of parameters, statistical properties of the task label space and hyperparameter settings. The general method may extend to the use of further a posteriori variables, for example training and inference times or confusion matrices.

A. Transfer Learning

Tasks: As source task for all experiments, we use ImageNet [45]. In our Transfer Learning processes, we used the following target tasks to calculate a knowledge transfer from the source task to the target task:

- t_1: CIFAR10 [46]
- t_2: MNIST [47]
- t_3: FashionMNIST [48]
- t_4: Places365 [49]
- $t_{5,6}$: Smartphones (original and augmented) [27]
- t_7: Hymenoptera (from a PyTorch [50] tutorial)

Models: We use the following pre-trained models, as implemented in PyTorch [50] and pre-trained on the source task, for our Transfer Learning processes:

- m_1: AlexNet [51], [52]
- m_2: VGG-16 [53]
- m_3: GoogLeNet [54]
- m_4: ResNet18 [55]

- m_5: SqueezeNet [56]
- m_6: DenseNet [57]
- m_7: ResNext [58]
- m_8: MobileNetV2 [59]
- m_9: Wide ResNet [60]
- m_{10}: ShuffleNet [61]
- m_{11}: MnasNet [62]

For each of these models, we examine the fine-tuning of the entire parameter vector and only of the classifier part, so that the models $m_{12:22}$ contain the classifier parts of the original models. If a model has no separate classifier part assigned to it, we use the last fully connected layer instead.

Transfer Learning dataset assembly: For our Meta Learning experiments, we assembled a dataset containing samples of Transfer Learning processes in various value ranges. The main part of our dataset consists of the task vector $\{t_1, t_2, t_3, t_5, t_7\}$, on which we performed Few-Shot Transfer Learning, using the optimization algorithms Adam [63], RMSProp [64] and plain stochastic gradient descent in all combinations with both cross entropy and negative log-likelihood loss as optimization criteria. We calculated each of these combinations with learning rates $\eta \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, number of epochs $n_e \in \{1, 2, 3\}$ and batch sizes $b \in \{25, 50\}$. This part of the dataset amounts to a total of $11,880$ Transfer Learning processes. For each of these processes, we also have saved the resulting neural network parameter vector, totaling in about $1.6TB$ of model data for future research. For another part of our Transfer Learning dataset, we collected metadata by fine-tuning each of our 22 vision models on the task vector $\{t_1, t_2, t_3, t_5, t_6, t_7\}$ with different learning rates $\eta \in \{1, 0.1, 0.01, 10^{-3}, 10^{-4}, 10^{-5}\}$ and number of epochs $e \in \{1, 3, 5, 7, 10\}$, but using only the Adam optimizer with cross entropy loss, resulting in a total of $3,960$ Transfer Learning processes. Furthermore, to have a barely transferable task, we added data from Transfer Learning processes to the hard target task t_4 (Places365), calculated with learning rates $\eta \in \{1, ..., 10^{-5}\}$ in a one-episode setting using the Adam optimizer and cross entropy loss, thus adding another 132 samples to our dataset.

B. Meta Learning

Metadata set: For each Transfer Learning process, we collected metadata containing both the settings of the Transfer Learning hyperparameters and the corresponding training and validation accuracies, as well as the corresponding training and testing time in seconds, the confusion matrices, and the memory usage on the device. The Transfer Learning hyperparameters include the loss function L, the optimizer o, the batch size b, the normalized number of epochs n_e and the normalized number of Transfer Learning model parameters n_p alongside their model identity m_i. From the task and model metadata, we also know the number of parameters and statistics about the task, such as the number of data points n_d, the number of classes n_c, and the stochastic moments of the label space $\mu_{1,...4}$. We divided this $15,972$ Transfer Learning process meta data sets into training and validation data and expanded it so that each of the tasks also poses a completely unseen test task, thus testing the generalization capability on data not included in the Meta Learning process. From the training meta data set, we sample batches of size 100 at random. We model only the a posteriori variable of validation accuracy in the target task, given different input configurations of the metamodel, but do so with numerous combinations of optimization algorithms, criteria and Meta Learning hyperparameters, as described in section IV. For a statistically meaningful study, we ran each Meta Learning trial ten times with different random initial weight vectors but used only the best meta-model from the ten trials for evaluation. To avoid overfitting in Meta Learning, we use early stopping.

Metamodel architectures: Our feedforward MLPs feature an output neuron Y, which tries to match the validation accuracy, with a sigmoid activation function. In the hidden layer, which we vary in our experiments as described in section IV, we employ $tanh()$ as activation function. The input layer includes the metadata as described by the various input configurations.

Input configuration representations: We represent optimization algorithm and the optimization criterion as one-hot encoded binary vectors, for example we encode our three optimization algorithms as:

$$o_i = \begin{cases} [0,0,1]: & Adam \\ [0,1,0]: & RMSProp \\ [1,0,0]: & SGD \end{cases} \quad (1)$$

In the same way, we proceed with the representation of the optimization criterion and each of our 22 model identities; for m_1 we set the first bit of the vector to 1, for model m_{22} the twenty-second bit. We normalize the numbers of model parameters, data points, classes, batch sizes and epochs across the maximum of all corresponding values in the training data set.

IV. Experiments

With our experiments we want to answer several questions; on the one hand we would like to know how good our method behaves compared to simple approaches. Since none of the known state of the art methods apply to our problem in this form, we compare the proposals of our method with simple approaches of linear regression and using the best Transfer Learning process known from the training set. On the other hand, we investigate the influences of different input data for the metamodels and address the question of whether there exist universally good parameters for our Meta Learning needs.

A. Experimental setups

We evaluated experimental setups using various input configurations with the a priori variables that we know of before the corresponding Transfer Learning process started in order

have a comparison to simply using the best-known model or using the approximation of a linear regression.

Common input variables: All input layer configurations c_i include the normalized batch size b, the normalized number of epochs n_e, the normalized number of Transfer Learning model parameters n_p. Also, each c_i contains and the learning rate η as a floating-point number as well as the loss function L, the optimizer o and the model identity m_i encoded as a one-hot binary vectors. We found that not including these variables lead to worse results, so we find a minimal working configuration in $c_0 = \{b, n_e, n_p, \eta, L, o, m_i\}$. For our investigations on the effect of stochastic moments of the label distributions, we denote the mean value of the source task as μ_{1_s} and the mean value of the target task as μ_{1_t}. Likewise, we denote the standard deviations as μ_{2_s}, μ_{2_t}, the skewness as μ_{3_s}, μ_{3_t} and the kurtosis as μ_{4_s} or μ_{4_t}. In short, we refer to the stochastic moments as $\mu_{1,2,3,4}$ and to the number of datapoints and classes as n_c and n_d, since we always use the source and target task values together.

Differences between input configurations: While configuration c_1 additionally contains the ratio of the number of classes and data points r_c and r_d between the source and target task, c_2 contains the number of data points and classes in source and target tasks n_{d_s}, n_{d_t}, n_{c_s}, n_{c_t} instead. Configuration c_3 features both representations of these quantities. We add kurtosis to c_2 to obtain c_4 and add skewness to c_4 to obtain c_5. Additionally, c_6 contains the ratios and c_7 the first stochastic moment, which we would interpret as an encoding for the task identity at this point. The configuration c_8 also takes the standard deviation into account, c_9 extends this list with the data and class ratios.

B. Results

Quality measure: We consider the average loss reductions of ten Meta Learning trials

$$\delta = 100\frac{L_{start}}{L_{end}} - 100[\%] \tag{2}$$

on the validation, training, and test metadata sets as a measure of the quality of our Meta Learning process. If the Meta Learning procedure had not reduced loss, so $L_{start} = L_{end}$, this would result in $\delta = 0\%$; with a loss reduction from $L_{start} = 1$ to $L_{end} = 0.5$, we would find $\delta = 100\%$. We chose the loss reduction rate δ instead of the mean average percentage error (MAPE) because we want to evaluate the Meta Learning processes themselves, independent of the loss function and metamodel performances.

Maximum loss reductions: First, we look at the maximum values of δ in table I and which metamodel configurations and hyperparameters for learning achieved them and find substantial loss reduction for certain configurations. For the training set, we find the metamodel configuration with the largest loss reduction in c_5. In detail, configuration c_5 achieves the maximum loss reduction $\delta_{train_{max}} = 219\%$ with a loss function $L = L2$, a learning rate $\eta = 10^{-3}$,

c	$\delta_{val_{max}}$	$\delta_{train_{max}}$	$\delta_{test_{max}}$
c_4	**188.12%**	200.55%	182.2%
c_5	162.07%	**209.71%**	**201.96%**
c_2	137.23%	138.37%	152.79%

TABLE I: Top three maximum loss reduction in percentages.

c	$\overline{\delta_{val}}$	$\overline{\delta_{train}}$	$\overline{\delta_{test}}$
c_4	**31.01%**	42.27%	10.87%
c_5	29.32%	**42.29%**	**11.56%**
c_2	29.32%	38.67%	9.31%

TABLE II: Top three mean loss reduction in percentages.

a number of epochs $n_e = 20$ and a number of hidden neurons $n_p = 100$ with the optimizer $o = RMSProp$ on a training set consisting of the tasks Hymenoptera, MNIST, FashionMNIST, Smartphones (original and augmented) and Places365. For the validation metadata set, we find an optimum at $\delta_{val_{max}} = 177\%$ with the input configuration $c = c_4$, the Meta Learning hyperparameters $\{L = L2, \eta = 10^{-3}, n_e = 30, n_p = 100, o = RMSProp\}$, as validated on metadata from the MNIST, Hymenoptera, Smartphones (augmented and original), Places365 and FashionMNIST tasks. When looking for the maximum loss reduction on the test datasets, the metamodel configuration $c = c_5$ achieved a loss reduction $\delta_{val_{max}} = 206\%$ with $\{L = L2, \eta = 10^{-3}, n_e = 20, n_p = 100, o = RMSprop\}$ for the FashionMNIST test task and the training/validation datasets with CIFAR10, MNIST, Hymenoptera smartphones (augmented and original) and Places365. From that observation we conclude that our Meta Learning processes can best assess the characteristics of a Transfer Learning process to the FashionMNIST target task, given the experimental setup as presented before.

Average loss reductions: Although we recommend the metamodels with the maximum values in the loss reduction for estimating the parameters for Transfer Learning and use them later for evaluation, we still try to answer the question which hyperparameters lead to a successful Meta Learning process. In the search for a Meta Learning process with good generalization ability, we do not look at maximum values, but for robust hyperparameters that perform well in all combinations of training, validation, and testing datasets. In that sense, table II confirms above average performances of the input configurations c_4 and c_5. This suggests that the normalized number of classes and data points may lead to better predictions than using the corresponding ratios. Furthermore, we see that using the third and fourth stochastic moments as input variables has a positive effect on the mean loss reduction, but that the first and second stochastic moments seem to distort the learning process, since we actually find loss increases in c_7, c_8 and c_9 when using mean and standard deviation as inputs. We find this rather counterintuitive, as we suspected that the label means would implicitly encode the tasks in such a way that metamodels could become absurdly accurate, but instead we found that

L	o	$\overline{\delta_{val}}$	$\overline{\delta_{train}}$	$\overline{\delta_{test}}$
L2	RMSProp	**31.82%**	**46.29%**	6.92%
L2	SGD	24.12%	30.48%	**8.23%**
L1	RMSProp	18.45%	25.02%	1.67%

TABLE III: Top three mean loss reductions in percentages for meta model optimizers.

n_p	n_e	η	$\overline{\delta_{val}}$	$\overline{\delta_{train}}$	$\overline{\delta_{test}}$
100	30	10^{-4}	**30.8%**	**41.9%**	**10.9%**
100	20	10^{-4}	28.4%	38.3%	10.6%
500	30	10^{-4}	24.6%	35.7%	4.4%

TABLE IV: Top three mean loss reductions in percentages for meta model architectures.

too large an input value negatively affects gradient descent.

Universally good parameters: We detail the most promising Meta Learning hyperparameters and architectures in tables III and IV to find universally useful Meta Learning settings. Table III shows, that the optimization criterion $L = L2$ and the optimization algorithm $o = RMSProp$ achieve the highest scores, averaged over all training and validation trials. To our surprise, the optimization algorithm $o = SGD$ with the optimization criterion $L = L2$ achieves the highest mean values on the test dataset. We suspect this to have happened by chance and would not assign statistical significance to the small difference, given the significantly worse performance on the training and validation data. In further search of what neural architectures give rise to metamodels with high generalization power, we can see in table IV the best results on the training, validation and test data set for 100 hidden neurons with a training time of 30 epochs at a learning rate of 10^{-4}, averaged over optimization criteria, algorithms and input configurations. We note with interest that increasing the number of neurons in the hidden layer of the metamodel does not lead to higher learning successes, since metamodels with 100 neurons already represent the facts in the best possible way.

C. Test set evaluations

Test setup: To check the validity of our metamodels and compare them to the aforementioned baselines, we choose an easy test task (FashionMNIST) and a hard test task (Places365) to query the metamodels for a one-episode Transfer Learning setting. We setup the training data such that the metamodels we use for inference had not seen any data from the respective test task during training or. We present the solutions that have the highest loss reduction on the test data set and derive the estimated accuracy for learning rates $\eta \in \{0.1, 10^{-3}, 10^{-4}, 10^{-5}\}$, for all models $m_{1:22}$, all optimization algorithms and optimization criteria, given the required and previously known metadata, such as the skewness and kurtosis of the label space, the normalized number of classes and data or the class and data ratios respectively. As we want to infer parameters for a one-episode Transfer

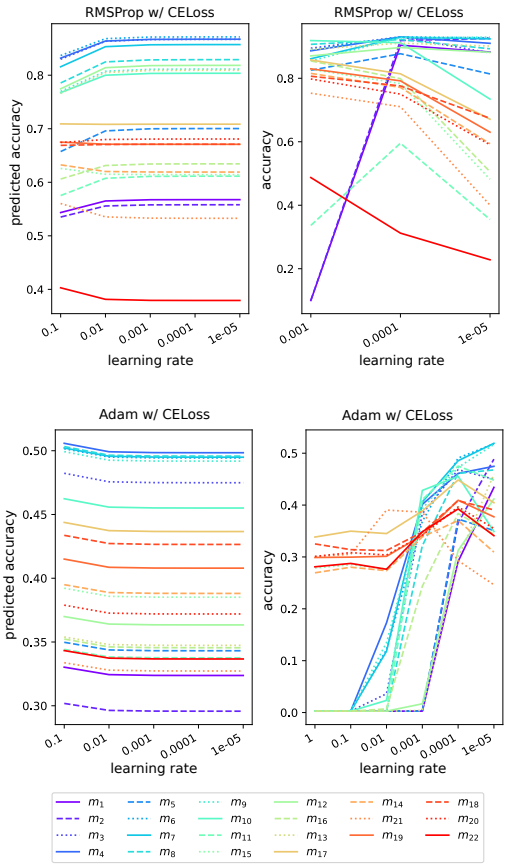

Fig. 2: Metamodel inferences (left) compared with the actual model accuracies (right) on the corresponding test tasks data from FashionMNIST (top) and Places365 (bottom).

Learning process, we set $n_e = \frac{1}{10}$, as we had a maximum value of ten for the number of epochs in the training data and we want to find out the performance after one episode. We set the normalized batch size value to $b = 1.0$ and compare the predictions of the metamodel with the actual Transfer Learning results from the test data set, as shown in Figure 2.

Evaluation of estimated accuracies: In the corresponding graphs in Figure 2, we see that the metamodel differentiates the expected accuracy in terms of learning rates and model identities for the easy task better than for the difficult task. Furthermore, we recognize smooth trajectories, which we consider as confirmation of Meta Learning success, since no random jumps occur in the predictions. This shows a successful metalearning process, but on the other hand, it also shows that the prediction quality decreases as the target task becomes more complex. Comparing the predicted and actual accuracies directly for a choice of optimizer and criterion, we see in Figure 2 that the metamodel could reproduce the model accuracies about the learning rates for the easy task in a practically useful way; i.e., for the trajectories m_1 and m_2 compared to the trajectory m_{22}. The hard task contains a jump at a learning rate of $\eta = 10^{-3}$, which the metamodel

rank	*acc*	m_i	η	o	L
1	93.7%	m_6	10^{-5}	*Adam*	*CE*
...					
24	**92.6%**	$\mathbf{m_2}$	$\mathbf{10^{-5}}$	**RMS**	**CE**
46	91.0%	m_4	10^{-5}	*RMS*	*CE*
58	90.3%	m_{13}	10^{-5}	*Adam*	*CE*
...					
537	7.7%	m_{10}	10^{-4}	*SGD*	*CE*

TABLE V: Actual test results for the FashionMNIST task, ranked by accuracy. Our method in bold letters, the best-known approach in blue and the linear regression proposal in teal.

rank	*acc*	m_i	η	o	L
1	51.91%	m_7	10^{-5}	*Adam*	*CE*
...					
7	**47.5%**	$\mathbf{m_4}$	$\mathbf{10^{-5}}$	**Adam**	**CE**
13	45.95%	m_{13}	10^{-5}	*Adam*	*CE*
85	17.2%	m_4	10^{-4}	*Adam*	*CE*
...					
141	0.26%	m_{10}	1.0	*Adam*	*CE*

TABLE VI: Actual test results for the Places365 task, ranked by accuracy.

could not anticipate.

Baseline comparison: To compare our Transfer Meta Learning method with the straightforward usage of the best setting known from the training metadata set and a proposal by linear regression as trained on the training metadata set as well, tables V and VI list the test results for all optimization algorithms and criteria. For a fair comparison, the linear regression baseline incorporates the same input variables as configuration c_5. As a straightforward baseline, we simply use the best configuration that we found in the training data set. Both baselines do not know data from the test set. In table V we can see that the metamodel has indeed proposed a decent configuration, which resulted in a higher test rank than we find for the baselines. The outcome for the hard task in table VI confirms the advantage of using a MLP prediction instead of a linear regression or simply the best-known approach, as it also performed better than the baselines in this case. We would reason that the characteristics of the hard task made it difficult to estimate Transfer Learning settings this task properly from the rest of the dataset and account for this behavior with an insufficient reflection of the necessary values in the training data.

V. Discussion

A. Limitations

Initially, our system underlies the same vulnerabilities as other Deep Learning systems and, if trained with harmful metadata, would propose inefficient Transfer Learning methods that reflect the training data. The limitations of our method lie in universal function approximation general or Deep Learning in particular. Especially in applications involving Continuous Learning, this general weakness can lead to a self-reinforcing problem. From information theoretic considerations, we find that the upper bounds as elaborated in [44] also apply to our method. We see the small number of target domain tasks as a limitation of the experiments we presented, considering that we only used one source task. Although the few task combinations already produced numerous Transfer Learning processes, more source and target tasks would certainly improve our Meta Learning processes. Another limitation of our experiments lies in that we only predict accuracy on the validation set of the target task. While we also recorded accuracy on the training set, learning times, inference times and memory usage, we leave the examination of the prediction of these values to future research.

B. Future work

First, we would like to state that in our next articles we will explore the applicability of our Transfer Meta Learning method to related learning problems, such as image segmentation [65] or image reconstruction [66], as well as to visual Reinforcement Learning [67]. In addition, we collected training accuracies, confusion matrices, resource utilizations, and timing information from our Transfer Learning processes and plan to explore the impact of predicting these data in terms of additional losses. We also ask to what extent the application of the Meta Learning method to the metadata created by Meta Learning could help our approach to make more accurate predictions. We will also include Transfer Learning methods other than fine-tuning, such as Elastic Weight Consolidation [24], Incremental Moment Matching [25] or Progressive Neural Networks [16]. These future studies would then include other types of Neural Networks, such as Variational Autoencoders, Long Short-Term Memory or Generative Adversarial Networks. After examining all of these effects and methods in our Transfer Meta Learning approach, we plan to conduct a baseline study with other benchmarks, such as VTAB [40], Causalworld [68], or cardiovascular disease recognition [69].

C. Conclusion

We proposed a new approach to facilitate Transfer Learning through Meta Learning with image classification as an example. Our results show that metamodels help to reduce computational cost by proposing appropriate settings for a Transfer Learning process for a target task. In particular, our approach only needs metadata and no actual samples from the dataset to propose Transfer Learning settings better than simply using the best-known settings or inferring settings by linear regression. Regarding the cost allocation for metadata collection, we argue that we initially performed these assessments for another study and then expanded them for demonstration purposes; the value of our method lies in the systematic reuse of all metadata that arises anyway. We think that further research may prove worthwhile, and we plan to pursue various leads as future work. We published our source code and data set under [70].

References

[1] N. Abou Baker, N. Zengeler, and U. Handmann, "A transfer learning evaluation of deep neural networks for image classification," *Machine Learning and Knowledge Extraction*, vol. 4, no. 1, pp. 22–41, 2022. 1, 2

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowl. and Data Eng.*, vol. 22, pp. 1345–1359, Oct. 2010. 2

[3] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016. 2

[4] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020. 2

[5] M. Kaboli, "A Review of Transfer LearningAlgorithms," research report, Technische Universität München, Aug. 2017. Transfer Learning Algorithms. 2

[6] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018. 2

[7] Y. Wang and Q. Yao, "Few-shot learning: A survey," *CoRR*, vol. abs/1904.05046, 2019. 2

[8] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 3987–3995, PMLR, 06–11 Aug 2017. 2

[9] F. Zenke, B. Poole, and S. Ganguli, "Improved multitask learning through synaptic intelligence," *CoRR*, vol. abs/1703.04200, 2017. 2

[10] V. J. Prakash and L. M. Nithya, "A survey on semi-supervised learning techniques," *CoRR*, vol. abs/1402.4645, 2014. 2

[11] X. Zhu, "Semi-supervised learning literature survey," 2006. 2

[12] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. 2

[13] I. Radosavovic, P. Dollár, R. B. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," *CVPR*, 2018. 2

[14] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *International Conference on Learning Representations*, 2019. 2

[15] I. Higgins, A. Pal, A. Rusu, L. Matthey, C. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner, "DARLA: Improving zero-shot transfer in reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 1480–1490, PMLR, 06–11 Aug 2017. 2

[16] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," *CoRR*, vol. abs/1610.04286, 2016. 2, 6

[17] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *CoRR*, vol. abs/1606.04671, 2016. 2

[18] M. Gómez-Silva, E. Izquierdo, A. de la Escalera, and J. M. Armingol, "Transferring learning from multi-person tracking to person re-identification," *Integrated Computer-Aided Engineering*, pp. 1–16, 04 2019. 2

[19] H. Chen, Y. Wang, Y. Shi, K. Yan, M. Geng, Y. Tian, and T. Xiang, "Deep transfer learning for person re-identification," pp. 1–5, 09 2018. 2

[20] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 1252–1260, Curran Associates, Inc., 2015. 2

[21] G. F. Elsayed, I. J. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," *CoRR*, vol. abs/1806.11146, 2018. 2

[22] S. Ruder, *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019. 2

[23] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2017. 2

[24] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016. 2, 6

[25] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4652–4662, Curran Associates, Inc., 2017. 2, 6

[26] M. Ghoreishi and A. Happonen, "Key enablers for deploying artificial intelligence for circular economy embracing sustainable product design: Three case studies," in *AIP Conference Proceedings*, vol. 2233, p. 050008, AIP Publishing LLC, 2020. 2

[27] N. Abou Baker, P. Szabo-Müller, and U. Handmann, "Feature-fusion transfer learning method as a basis to support automated smartphone recycling in a circular smart city," in *EAI S-CUBE 2020 - 11th EAI International Conference on Sensor Systems and Software*, (online), 2020. 2

[28] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. R Oleiwi, "Towards a better understanding of transfer learning for medical imaging: a case study," *Applied Sciences*, vol. 10, no. 13, p. 4523, 2020. 2

[29] J. Shen, X. Xiong, Y. Li, W. He, P. Li, and X. Zheng, "Detecting safety helmet wearing on construction sites with bounding-box regression and deep transfer learning," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 2, pp. 180–196, 2021. 2

[30] Y. Guo, H. Niu, and S. Li, "Safety monitoring in construction site based on unmanned aerial vehicle platform with computer vision using transfer learning techniques," in *Proceedings of the 7th Asia-Pacific Workshop on Structural Health Monitoring, APWSHM 2018, 12-15 November 2018, Hong Kong SAR, China*, 2018. 2

[31] J. Vanschoren, "Meta-learning: A survey," *CoRR*, vol. abs/1810.03548, 2018. 2

[32] H. Latapie, O. Kilic, G. Liu, Y. Yan, R. Kompella, P. Wang, K. R. Thorisson, A. Lawrence, Y. Sun, and J. Srinivasa, "A metamodel and framework for artificial general intelligence from theory to practice." 2

[33] Y.-C. Chuang, T. Chen, Y. Yao, and D. S. H. Wong, "Transfer learning for efficient meta-modeling of process simulations," *Chemical Engineering Research and Design*, vol. 138, pp. 546–553, 2018. 2

[34] M. Ashouri and A. Hashemi, "A transfer learning metamodel using artificial neural networks applied to natural convection flows in enclosures," 2020. 2

[35] D. J. Skinner and R. Maulik, "Meta-modeling strategy for data-driven forecasting." 2

[36] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[37] Q. Sun, Y. Liu, Z. Chen, T.-S. Chua, and B. Schiele, "Meta-transfer learning through hard tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. 2

[38] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[39] S.-J. Park, S. Han, J.-W. Baek, I. Kim, J. Song, H. B. Lee, J.-J. Han, and S. J. Hwang, "Meta variance transfer: Learning to augment from the others," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 7510–7520, PMLR, 13–18 Jul 2020. 2

[40] X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, L. Beyer, O. Bachem, M. Tschannen, M. Michalski, O. Bousquet, S. Gelly, and N. Houlsby, "The visual task adaptation benchmark," *CoRR*, vol. abs/1910.04867, 2019. 2, 6

[41] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer learning and meta learning-based fast downlink beamforming adaptation," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1742–1755, 2020. 2

[42] F. Aiolli, "Transfer learning by kernel meta-learning," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 81–95, JMLR Workshop and Conference Proceedings, 2012. 2

[43] P. Kordík, J. Černỳ, and T. Frỳda, "Discovering predictive ensembles for transfer learning and meta-learning," *Machine learning*, vol. 107, no. 1, pp. 177–207, 2018. 2

[44] S. T. Jose, O. Simeone, and G. Durisi, "Transfer meta-learning: Information-theoretic bounds and information meta-risk minimization," *IEEE Transactions on Information Theory*, 2021. 2, 6

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009. 2

[46] R. Zaheer and H. Shaziya, "A study of the optimization algorithms in deep learning," pp. 536–539. 2

[47] O. Kaziha and T. Bonny, "A comparison of quantized convolutional and lstm recurrent neural network models using mnist," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1–5, IEEE, 11/19/2019 - 11/21/2019. 2

[48] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. 2

[49] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 2

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019. 2

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, eds., *ImageNet Classification with Deep Convolutional Neural Networks*, 2012. 2

[52] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *CoRR*, vol. abs/1404.5997, 2014. 2

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." 2

[54] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. 2

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." 2

[56] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size." 3

[57] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE, 7/21/2017 - 7/26/2017. 3

[58] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, IEEE, 7/21/2017 - 7/26/2017. 3

[59] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications." 3

[60] S. Zagoruyko and N. Komodakis, "Wide residual networks." 3

[61] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices." 3

[62] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. Le V, "Mnasnet: Platform-aware neural architecture search for mobile," *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR 2015*, 2015. 3

[64] T. Tieleman and G. Hinton, "Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude," 2012. 3

[65] S. Ghosh, N. Das, I. Das, and U. Maulik, "Understanding deep learning techniques for image segmentation," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–35, 2019. 6

[66] A. Raj, Y. Bresler, and B. Li, "Improving robustness of deep-learning-based image reconstruction," in *International Conference on Machine Learning*, pp. 7932–7942, PMLR, 2020. 6

[67] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, IEEE, 2016. 6

[68] O. Ahmed, F. Träuble, A. Goyal, A. Neitz, Y. Bengio, B. Schölkopf, M. Wüthrich, and S. Bauer, "Causalworld: A robotic manipulation benchmark for causal structure and transfer learning," *arXiv preprint arXiv:2010.04296*, 2020. 6

[69] M. Boulares, T. Alafif, and A. Barnawi, "Transfer learning benchmark for cardiovascular disease recognition," *IEEE Access*, vol. 8, pp. 109475–109491, 2020. 6

[70] "Source code and dataset." https://gitlab.hs-ruhrwest.de/nico.zengeler/transfer-meta-learning. Accessed: 2022-01-24. 6