Convex Base Learners in Few-shot Meta-Learning Benchmark

Shuailong Zhu [1] **Haochen Su** [1]

Abstract

Few-shot learning has become essential for producing models that generalize from a few examples. Among few-shot learning algorithms, many meta-learning approaches consist of simple base learners such as the nearest neighbor classifier. However, according to MetaOptNet, compared to the simple base learners, utilizing a linear predictor as the base learner enhances the performance of the model to learn embeddings for meta-learning. MetaOptNet also incorporates a metric scaling as TADAM does. It enables the model to learn the optimal similarity metric during the training. To efficiently solve the objective of the linear predictor, MetaOptNet exploits the dual formulation to use high-dimensional embeddings at a modest increase in computational overhead. We implement MetaOptNet for the few-shot benchmark(with Tabula Muris dataset and SwissProt dataset), compare it with existing algorithms including baseline, baseline with finetune(baseline_pp), MAML, MatchingNet and ProtoNet, and achieve better performance.

1. Introduction

Human beings can learn how to identify new categories from only few examples. Few-shot learning, cast as a meta-learning problem, has attracted significant attention (Vinyals et al., 2016; Snell et al., 2017), as it aims to produce models that can generalize from small amounts of labeled data. One aims to learn a model that extracts information from a set of support examples (sample set) to predict the labels of instances from a query set.

Two approaches have attracted significant attention in the metric-based methods of the few-shot learning domain: Matching Networks (Vinyals et al., 2016), and Prototypical Networks (Snell et al., 2017). The former represents each class by the mean embedding of the examples, and the classification rule is based on the distance to the nearest class mean, while prototypical networks that learns a kernel density estimate of the class densities using the embeddings over training data (similar to the soft attention mechanism). Since then, learning the best-adapted metric for few-shot learning has been the interest of the field. Learning a metric space in the context of few-shot learning generally implies identifying a suitable similarity measure, a feature extractor mapping raw inputs onto similarity space. MetaOptNet(Lee et al., 2019) uses the scaling metric as TADAM(Oreshkin et al., 2018) does. It is shown that adjusting the prediction score by a learnable scale parameter provides better performance for base learners like nearest neighbors and ridge regression etc.

More importantly, MetaOptNet focuses on the analysis of the role of linear classifiers in metric-based methods. The approach is shown in Figure 1, where a linear support vector machine (SVM) is utilized as the linear predictor. Thanks to the convexity as a quadratic program (QP), the minimization of the empirical generalization error (global minima) of the queries can be computed using gradient-based methods. To ensure the computational complexity, we can transfer this problem to its dual form through KKT and implicit function theorem. Specifically, we use the formulation of (Amos & Kolter, 2017) which provides efficient GPU routines for computing solutions to QPs and their gradients.

We implement MetaOptNet and compare the performance of it with other methods in the few-shot benchmark(with Tabula Muris dataset(Cao et al., 2020) and SwissProt dataset(swi)). From the experiments and our ablation study, it turns out the properties of MetaOptNet enhance the performance and make it achieve better results.

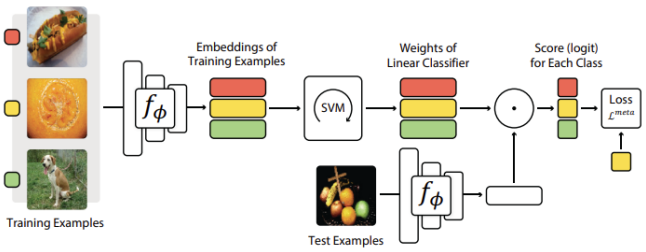

Figure 1. Pipeline of MetaOptNet(Lee et al., 2019)

2. Related Work

The study of few-shot classification especially using meta-learning has been of interest for some time. MAML(Finn et al., 2017), MatchingNet(Vinyals et al., 2016) and ProtoNet(Snell et al., 2017) are three important meta-learning algorithms that are included in the benchmark.

MAML

MAML(Finn et al., 2017) approach aims to meta-learn an initial condition (set of neural network weights) that is good for fine-tuning on few-shot problems. Their strategy is to search for a good weight initialization of a given neural network such that it can be effectively generalized and fine-tuned on a few-shot setting within a few gradient-descent update steps. The intuition behind this approach is that some internal representations are more transferrable than others. In effect, MAML method aims to optimize the model parameters such that one or a small number of gradient steps on a new task will produce maximally effective behavior on that task. In practice, MAML meta-gradient update involves a gradient through a gradient. We can understand it as copying an original model, computing gradient descent on the copy and do the update on the copy, then do the second order gradient computation on the copy but doing the update on the original model.

MatchingNet

MatchingNet(Vinyals et al., 2016) model architecture follows advances in neural networks augmented with memory that enables rapid learning. MatchingNet learns a kernel density estimate of the class densities using the embeddings over training data (the model can also be interpreted as a form of attention over training examples). In practive, after getting the embedding from the sampled data, they stack the embedding and feed them into a bidirectional Long-Short Term Memory (LSTM)(Vennerød et al., 2021) to do encoding. Cosine distance is computed to measure the similarity and classification is based on nearest neighbor matching.

ProtoNet

ProtoNet(Snell et al., 2017) uses a simpler architecture, it represents each class by the mean embedding of the examples, and the classification rule is also based on the distance to the nearest class mean. There are similarities between MatchingNet and ProtoNet. But MatchingNet compares cosine distance between the query feature and each support feature, and computes average cosine distance for each class, while ProtoNet compares the Euclidean distance between query features and the class mean of support features. It is claimed that Euclidean distance performs better than Cosine distance. ProtoNet is known for its simplicity and efficiency.

3. Method

We first introduce meta-learning pipeline and then discuss how convex base learners, such as linear SVMs.

3.1. Meta-Learning Pipeline

Assume we have an embedding model f_ϕ, a predictor f_θ, and a base learner $\mathcal{A}$, and a loss function $\mathcal{L}^{base}$ specified for base learner like linear predictor, the objective of the base learner is to compute the parameter of predictor based on an embedding model and the distribution of the data $\mathcal{D}$,

$$\theta = \mathcal{A}(\mathcal{D}; \phi) = \underset{\theta}{argmin}\, \mathcal{L}(\mathcal{D}; \theta, \phi) \tag{1}$$

Given a collection of tasks: $\mathcal{T} = \{(\mathcal{D}^{train}, \mathcal{D}_i^{test})\}_{i=1}^{I}$, where $\mathcal{T}_i$, consisting of a training (support) and a test dataset (query) often refers to a task. The objective is to learn an embedding model ϕ that minimizes the test error L^{error} across tasks given a base learner $\mathcal{A}$,

$$\min_{\phi} \mathbb{E}_{\mathcal{T}}[\mathcal{L}^{error}(\mathcal{D}^{test}; \theta, \phi)] \tag{2}$$

where $\theta = \mathcal{A}(\mathcal{D}^{train}; \phi)$.

Episode Training

In practice, our dataset do not explicitly contain tasks like $(\mathcal{D}^{train}, \mathcal{D}_i^{test})$, but each task is constructed dynamically during the meta-training stage, which is also defined as an episode. We denote the overall set of categories is C^{train}, and for each episode, we sample a set of categories C_i of size K (K-way). Then the training set (support set) $\mathcal{D}^{train_i} = \{(\mathbf{x}_n, y_n) | n = 1, ..., N \times K\, y_N \in C_i\}$ consisting of N images per category (N-shot) is sampled. $\mathcal{D}_i^{test}$ is sampled by similar process from the categories C_i.

3.2. Base Learners

SVM

Firstly we consider our base learner based on multi-class linear classifiers (SVMs, ridge regression). The k-class linear SVM can be written as

$$\theta = \mathcal{A}(D^{train}; \phi) = \underset{\{\mathbf{w}_k\}}{argmin} \min_{\{\xi_i\}} \frac{1}{2} \sum_k ||\mathbf{w}_k||_2^2 + C \sum_n \xi_n$$

subject to

$$\mathbf{w}_{y_n} \cdot f_\phi(\mathbf{x}_n) - \mathbf{w}_k \cdot f_\phi(\mathbf{x}_n) \geq 1 - \delta_{y_n,k} - \xi_n, \forall n, k \tag{3}$$

where $\theta = \{\mathbf{w}_k\}$, C is the regularization parameter.

The objective of the SVM is convex, differentiable, and has a unique optimum. We can also obtain the dual formulation.

Let

$$\mathbf{w}_k(\boldsymbol{\alpha}^k) = \sum_n \alpha_n^k f_\phi(\mathbf{x}_n), \forall k \tag{4}$$

The dual formulation is,

$$\begin{aligned} &\max_{\{\boldsymbol{\alpha}^k\}} [-\frac{1}{2}\sum_k ||\mathbf{w}_k(\alpha_k)||_2^2 + \sum_n \alpha_n^{y_n}] \\ &\text{subject to} \\ &\alpha_n^{y_n} \leq C, \alpha_n^k \leq 0, \forall k \neq y_n \\ &\sum_k \alpha_n^k = 0, \forall n \end{aligned} \tag{5}$$

The reason why we need to use the dual formulation is the size of the optimization variable is $N \times K$, which is often much smaller than the feature dimension. Thus, we solve the dual QP in our implementation.

Ridge Regression

(Bertinetto et al., 2018) utilized ridge regression with one-hot label as the base learner. The optimization is also a QP and can be implemented within the framework

$$\max_{\{\boldsymbol{\alpha}^k\}} [-\frac{1}{2}\sum_k ||\mathbf{w}_k(\alpha_k)||_2^2 + \sum_n \alpha_n^{y_n} - \frac{\lambda}{2}\sum_k ||\boldsymbol{\alpha}^k||_2^2] \tag{6}$$

The experiments in Section 4 highlights the advantage of SVM, which is consistent with the result in (Lee et al., 2019).

4. Experiments and Results

For the given five baseline methods(baseline, baseline_pp, MAML, MatchingNet, ProtoNet), we directly ran it with the default setting in the benchmark. For MetaOptNet(Lee et al., 2019), we did hyper-parameter tuning on the two datasets in the benchmark. Based on the **validation** accuracy, we selected the hyper-parameters and found out the hyper-parameters we chose are actually quite the same as the ones mentioned in their paper. In table 1, experiment results of test accuracy on the two datasets are shown. MetaOptNet's result after hyper-parameter tuning is shown.

In Table 1, MetaOptNet achieves best performance in Tabula Muris dataset and second place in SwissProt dataset. From the results we can find that linear classifiers indeed offer better generalization than nearest-neighbor classifiers at a modest increase in computational costs(exploiting dual formulation and KKT condition), and SVM base learner performs better than Ridge Regression in our classification task,which is also consistent with the findings in (Lee et al., 2019).

Table 1. Experiment results of test accuracy on the two datasets. The five baseline methods are in default setting. The MetaOptNet method is after parameter adjustment.

METHOD	TABULA MURIS	SWISSPROT
BASELINE	87.06± 8.23	66.35± 7.37
BASELINEPP	83.99± 9.11	57.49± 7.75
MAML	85.70± 8.89	45.06± 7.46
MATCHINGNET	81.76± 10.34	60.64± 8.05
PROTONET	86.39± 8.95	62.50± 7.66
METAOPTNET(RIDGE)	86.86± 8.29	64.30± 7.34
METAOPTNET(SVM)	87.90± 8.16	64.40± 7.40

Implementation Details and Hyper-parameter Tuning for MetaOptNet

Since all methods have an embedding network(backbone), and what we focus more is on the added components and properties of those components in MetaOptNet, we choose not to change the dataset configuration(including backbone). So what we used as backbone for each dataset is Fully Connected Network with layer dimensions [64,64] for Tabula Muris and Fully Connected Network with layer dimensions [512,512] for SwissProt.

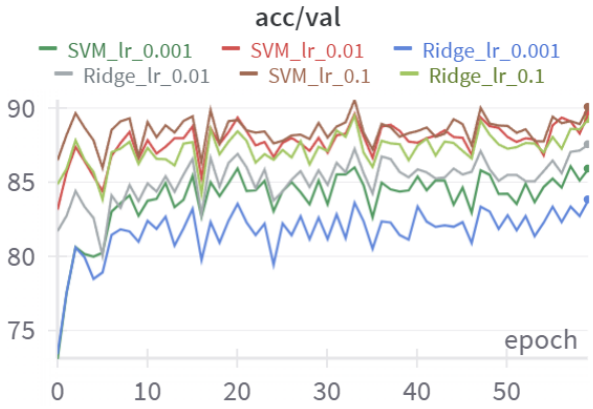

Figure 2. Experiments on learning rate and base learner type on Tabula Muris dataset

We followed the setting of the optimizer from the original paper (SGD, $lr = 0.1$, $weight\_decay = 0.0005$, $epoch = 60$) and did experiments on multiple initial learning rates $(0.1, 0.01, 0.001)$ for SVM base learner and Ridge Regression base learner for hyper-tuning for both our datasets. As stated in (Lee et al., 2019), the initial learning rate set to be 0.1, combined with a learning rate scheduler (0.006, 0.0012, and 0.00024 at epochs 20, 40 and 50) indeed makes the model converge faster and achieve better **validation accuracy** as shown in Figure 2 (For readability, only

Tabula Muris dataset's results are shown). Both training loss and validation accuracy are steady and achieve their best performance after 60 epochs.

5. Ablation Study

Our ablation study mainly focus on two aspects. The comparison of the base learners SVM and Ridge Regression, and the effect of the learnable scale factor.

SVM or Ridge Regression

From our experiment result shown in Figure 2 and Table 1, it is quite clear that SVM performs better than Ridge Regression as base learner in our task. As stated in (Lee et al., 2019), ridge regression may not be best suited for classification problems. It still can achieve reasonably good results in practice by training models by minimizing squared error with respect to one-hot labels. But indeed, it does not perform as good as SVM in our classification task, which also matches the results in (Lee et al., 2019).

Effect of Learnable Scale Factor

Prior work in few-shot learning (Oreshkin et al., 2018; Lee et al., 2019) suggest that adjusting the prediction score by a learnable scale parameter is beneficial for the meta-learning with SVM and ridge regression base learner. We investigate whether this is true on our two datasets. As shown in Figure 3 and 4 and Table 2(for readability and limit of max page number, only results on Tabula Muris are shown), adding the learnable scale parameter indeed enhances the performance of the model. It makes the training achieve lower loss and makes the convergence faster. It also enhances the validation and test accuracy. With the empirical result, we reach a conclusion that the scale factor indeed enables the model to learn the optimal similarity metric during the training, therefore contributing to the overall model performance.

Table 2. Experiments results(test accuracy) on effect of the learnable scale factor, all the other parameters are fine-tuned for each case

METAOPTNET	TEST ACCURACY
RIDGE(NO SCALE)	85.43±9.01
RIDGE(SCALED)	86.86± 8.29
SVM(NO SCALE)	79.22±10.48
SVM(SCALED)	87.90± 8.16

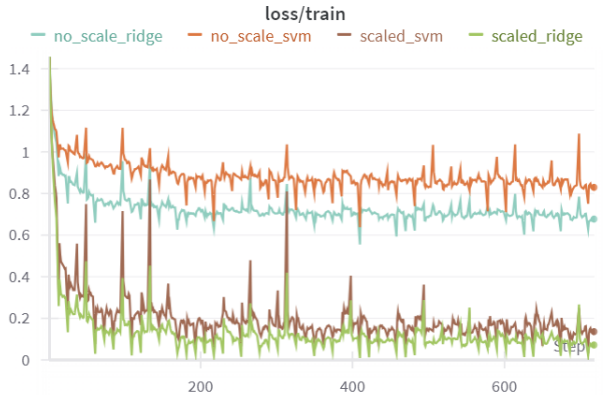

Figure 3. Experiments on effect of the learnable scale factor, all the other parameters are fine-tuned for each case

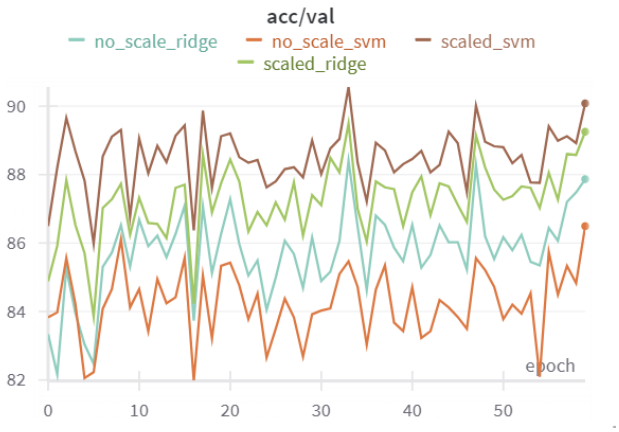

Figure 4. Experiments on effect of the learnable scale factor, all the other parameters are fine-tuned for each case

6. Conclusion

In this project, we implement MetaOptNet(Lee et al., 2019), and compare it with other baseline algorithms in the benchmark after fine-tuning it. MetaOptNet is a meta-learning approach with convex base learners for few-shot learning. It turns out that our new implemented algorithm performs the best among all the meta-learning algorithms. Consistent with the result in (Lee et al., 2019), we can conclude that convex classifiers indeed offer better generalization than nearest-neighbor classifiers. We further investigate that the components in MetaOptNet(SVM base learner and learnable scaled metric) do contribute to its performance.

References

Uniprot. URL https://www.uniprot.org/.

Amos, B. and Kolter, J. Z. Optnet: Differentiable optimization as a layer in neural networks. *CoRR*, abs/1703.00443, 2017. URL http://arxiv.org/abs/1703.00443.

Bertinetto, L., Henriques, J. F., Torr, P. H. S., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. *ArXiv*, abs/1805.08136, 2018. URL https://api.semanticscholar.org/CorpusID:29153681.

Cao, K., Brbic, M., and Leskovec, J. Concept learners for generalizable few-shot learning. *CoRR*, abs/2007.07375, 2020. URL https://arxiv.org/abs/2007.07375.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. volume abs/1703.03400, 2017. URL http://arxiv.org/abs/1703.03400.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

Oreshkin, B., Rodríguez López, P., and Lacoste, A. Tadam: Task dependent adaptive metric for improved few-shot learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/66808e327dc79d135ba18e051673d906-Paper.pdf.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Vennerød, C. B., Kjærran, A., and Bugge, E. S. Long short-term memory RNN. *CoRR*, abs/2105.06756, 2021. URL https://arxiv.org/abs/2105.06756.

Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., and Wierstra, D. Matching networks for one shot learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.