ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Probabilistic Adaptation for Meta-Learning

Tameem Adel[1] *

1- National Physical Laboratory - Data Science Department
Hampton Rd, Teddington - United Kingdom

Abstract. Meta-learning models learn to generalise to unseen tasks at test time. We introduce a meta-learning algorithm which balances (global) generalisation with a (local) adaptive mechanism allowing the meta-learner to deal with potentially substantial heterogeneity in the task distribution. The proposed meta-learner flexibly consolidates shared components (responsible for generalisation) with task-specific components. The latter components are adapted, in a data-driven manner, based on estimating the similarity between the meta-test task in hand and the training tasks. Experiments demonstrate improved performance on few-shot learning benchmarks, both general and others involving a more heterogeneous set of tasks.

1 Introduction

One of the distinguishing features of human intelligence is the ability to understand new concepts, something which machine learning algorithms aim to emulate [1]. Meta-learning (ML), or learning-lo-learn [2, 3, 4, 5], is a machine learning paradigm addressing these issues. ML algorithms learn from data belonging to some tasks in order to perform on other tasks which were unobserved during meta-training.

To better understand the problem in hand, humans are adept at implicitly assessing similarities with previously encountered tasks. A human asked to go from point A to B in a certain city can assess which characteristics of the roads are city-specific and which are salient across cities. Most of the current ML algorithms generically establish a single ML model without leveraging any adaptation to the meta-test task in hand.

We propose an ML algorithm which aims at getting the best of both worlds: It globally learns a general ML model, while being (locally) adaptive to every meta-test task. Rather than being limited to rigidly applying a pre-specified number of gradient steps to the ML model when encountering a meta-test task, the proposed ML algorithm can as well adapt its task learning by estimating the similarity between the training tasks and the current meta-test task. Similarities are compared over a low-dimensional space.

We perform experiments on three few-shot learning (FSL) benchmarks, achieving state-of-the-art (SOTA) results in general FSL settings as well as in settings involving more heterogeneity in the task distribution.

To summarise, our principal contributions can be described as follows:

- We propose the probabilistic adaptation for meta-learning (`PAML`) algorithm which can address high levels of heterogeneity in the task distribution (Section 2).
- The proposed `PAML` efficiently adapts the ML optimisation to each meta-test task.
- State-of-the-art results on three FSL benchmarks demonstrate the efficacy of `PAML` to learn from heterogeneous sets of tasks (Section 3).

*Work mostly done while at the University of Cambridge, Machine Learning Group (MLG), UK.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

2 Probabilistic Adaptation for Meta-Learning (PAML)

In meta-learning (ML), the generalisation power is key to succeed in learning from unseen tasks. Yet, the prospect of learning a single ML model which is optimal for every unseen task can be unrealistic (no free lunch). To that end, we introduce the Probabilistic Adaptation for Meta-Learning (PAML) algorithm, which efficiently automates a (local) adaptation procedure for each meta-test task based on the most similar training task.

Notation and problem setup. Let $\mathcal{X}$ and $\mathcal{Y}$ refer to the input and output spaces, respectively, and let $\mathbf{X}$ and $\mathbf{Y}$ refer to the input and output variables. A task $\mathcal{T}$ is characterised by a data distribution $\mathcal{D}$ on $\mathcal{X}$, and a labeling function f. A hypothesis space $\mathcal{H}$ is a space of functions h. Under a distribution $\mathcal{D}_{\mathcal{T}}$, the error of a hypothesis h w.r.t. the labeling function $f_{\mathcal{T}}$ is depicted by: $\varepsilon_{\mathcal{T}}(h, f_{\mathcal{T}}) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{T}}}[|h(\mathbf{x}) - f_{\mathcal{T}}(\mathbf{x})|]$. We address a few-shot (K-shot) meta-learning setup where the model is trained on T tasks drawn from a task distribution $\Pr(\mathcal{T})$. During meta-training, two samples are drawn from each training task $\mathcal{T}_i$. The first sample contains K points per class $S^{t_{tr}} = (\{\mathbf{x}_i^{t_{tr}}, \mathbf{y}_i^{t_{tr}})\}_{i=1}^{N_t}$ [1], and there is another sample $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}})\}_{i=1}^{M_t}$, where $t \in 1, \ldots, T$ is the (training) task index, N_t and M_t are the two sample sizes of task t. For every meta-test task, solely $S^{t_{tr}}$ is observed whereas $S^{t_{ts}}$ is used for evaluation.

2.1 The PAML Model

The PAML model involves shared parameters and task-specific parameters. Updating the shared parameters depends on all the tasks. The task-specific parameters are adapted based on the most similar training task. The level of involvement of the adaptation in the optimisation procedure is learnt in a data-driven manner.

Let tasks $\mathcal{T}_1, \ldots, \mathcal{T}_T$ refer to the training tasks, and refer to the current test task as $\mathcal{T}_z$, with task-specific parameters θ^z. Denote by α the shared parameters. The most similar task to $\mathcal{T}_z$ is determined out of the training task-specific parameters $\theta^1, \ldots, \theta^T$.

2.2 Parameter Learning

The first step in updating the task-specific parameters θ^z is conceptually similar to previous ML frameworks, e.g. MAML [6]. Starting from an initial condition θ, a few gradient steps are taken based on the training sample $S^{t_{tr}}$:

$$\theta^z = \theta - \gamma_1 \nabla_\theta \varepsilon_z(h_{\theta^z}, f_z) \tag{1}$$

where the hyperparameters γ_1, γ_2 and γ_3 (the latter two are shown later) are learning rates. The number of the gradient steps (we show one for simplicity) is another hyperparameter.

The second phase is to adapt θ^z based on the most similar training task $\mathcal{T}^{\text{NN}}$. Due to the first phase in (1), the learner has already become informed about $\mathcal{T}_z$, which enables it to search for the most similar training task. To determine $\mathcal{T}_{\text{NN}}$, the classification output (probabilities per class) of θ^z on the sample $S^{t_{tr}}$ of task $\mathcal{T}_z$ is compared to the corresponding output of $\theta^1, \ldots, \theta^T$ on the same sample $S^{t_{tr}}$. This is done by placing a softmax layer denoting classes of the current task $\mathcal{T}_z$ on top of $\theta^1, \ldots, \theta^T$.

[1] $N_t = K \times$number of classes.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

A total of T pairwise comparisons of the probabilistic outputs between $\mathcal{T}_z$ and $\mathcal{T}^1, \ldots, \mathcal{T}^T$ are performed using the cosine similarity. The most similar task $\mathcal{T}_{\text{NN}}$ is the one achieving the highest similarity to $\mathcal{T}_z$. Refer to the cosine similarity as sim, to the dot-product as $\cdot$, and to a task output on $S^{t_{tr}}$ as $f_{\mathcal{T}}(S^{t_{tr}})$:

$$\text{sim}(z, j) = \frac{f_z(S^{t_{tr}}) \cdot f_j(S^{t_{tr}})}{\|f_z(S^{t_{tr}})\| \|f_j(S^{t_{tr}})\|}, \quad j = 1, 2, \ldots, T \tag{2}$$

The index of the most similar task $\mathcal{T}_{\text{NN}}$ is the one maximising (2):

$$\text{NN} = \text{argmax}_j \ \text{sim}(z, j), \quad j = 1, 2, \ldots, T \tag{3}$$

Hence, the performance on the sample $S^{t_{tr}}$ of task $\mathcal{T}_z$ is used as a proxy to estimate the similarity between the $\mathcal{T}^z$ and the training tasks. This proxy has its theoretical grounding [7]. The developed proxy is also sound from the efficiency perspective since the output space is considerably smaller than the parameter space.

Using a part of the sample $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}}\})_{i=1}^{M_z/2}$ of task $\mathcal{T}_z$, the second update to the parameters θ^z is then performed by taking a few gradient steps (one step is next shown for simplicity). The value of θ^z obtained from (1) is further updated as follows:

$$\theta^z = \theta^z - \gamma_2 \nabla_\theta \varepsilon_z(h_{\theta^z}, f_z) \ + \frac{1}{1 + \text{e}^{-\lambda}}(\theta^{\text{NN}} - \theta^z), \ \ \lambda = \varepsilon_z(h_{\theta^z}, f_z) - \varepsilon_{\text{NN}}(h_{\theta^{\text{NN}}}, f_{\text{NN}}) \tag{4}$$

The adaptation spectrum, controlled by λ, moves along these extremes: For a data point, if the error of task $\mathcal{T}_z$, $\varepsilon_z(h_{\theta^z}, f_z)$, is considerably larger than the corresponding error of the most similar task, $\varepsilon_{\text{NN}}(h_{\theta^{\text{NN}}}, f_{\text{NN}})$, then λ is large enough to allow the task-specific parameters θ^z to be adapted based on (the more accurate) θ^{NN}. In the opposite case, λ will be close to zero and in such case the adaptation impact will be negligible.

The task-specific parameters of all the sampled tasks are used to update the shared parameters α by taking gradients on parts from the samples $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}}\})_{i=M_t/2+1}^{M_t}$:

$$\alpha = \alpha - \gamma_3 \sum_{\mathcal{T} \sim Pr(\mathcal{T})} \nabla_\alpha \, \varepsilon_{\mathcal{T}}(h_{\theta\mathcal{T}}, f_{\mathcal{T}}) \tag{5}$$

The key steps of the optimisation procedure are listed in Algorithm 1.

3 Experiments

To empirically evaluate the effectiveness of the proposed ML framework, `PAML`, we compare it to various SOTA ML algorithms on FSL tasks. We chiefly aim at evaluating the following aspects: 1) Classification performance on general FSL benchmarks (Omniglot and mini-ImageNet); and 2) Classification performance on more challenging settings in which tasks are more heterogeneous (OVD).

We conduct our experiments on the following FSL datasets: Omniglot [8], mini-ImageNet [9] and the multi-dataset OVD [4]. We experiment on K-shot C-way tasks, where K stands for the number of data examples per class, and C stands for the number

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Algorithm 1 Probabilistic Adaptation for Meta-Learning (PAML)

Input: A task distribution $Pr(\mathcal{T})$, from which T meta-training tasks were sampled, each with two data samples: $S^{t_{tr}} = (\{\mathbf{x}_i^{t_{tr}}, \mathbf{y}_i^{t_{tr}})\}_{i=1}^{N_t}$ and $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}})\}_{i=1}^{M_t}$.
Input: γ_1, γ_2, γ_3: Learning rate hyperparameters.
Randomly initialise θ
while not done **do**
for a batch of sampled tasks **do**
Process one of the sampled tasks $\mathcal{T}_z \sim Pr(\mathcal{T})$
Draw two samples $S^{t_{tr}}$ and $S^{t_{ts}}$ from $\mathcal{T}_z$.
Evaluate $\nabla_\theta \varepsilon_z(h_{\theta^z}, f_z)$ with respect to the data points of the sample $S^{t_{tr}}$.
Phase 1: Update parameters θ^z with gradient descent: $\theta^z = \theta - \gamma_1 \nabla_\theta \varepsilon_z(h_{\theta^z}, f_z)$
Learn the most similar task to $\mathcal{T}_z$, referred to as $\mathcal{T}_{\text{NN}}$, via (2) and (3).
Evaluate $\varepsilon_z(h_{\theta^z}, f_z)$ w.r.t. the first half of $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}})\}_{i=1}^{M_z/2}$.
Evaluate $\varepsilon_{\text{NN}}(h_{\theta^{\text{NN}}}, f_{\text{NN}})$ w.r.t. $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}})\}_{i=1}^{M_z/2}$.
Compute $\lambda = \varepsilon_z(h_{\theta^z}, f_z) - \varepsilon_{\text{NN}}(h_{\theta^{\text{NN}}}, f_{\text{NN}})$
Evaluate $\nabla_\theta \varepsilon_z(h_{\theta^z}, f_z)$ w.r.t. the first half of $S^{t_{ts}} = (\{\mathbf{x}_i^{t_{ts}}, \mathbf{y}_i^{t_{ts}})\}_{i=1}^{M_z/2}$.
Phase 2: Adapt θ^z using: $\theta^z = \theta^z - \gamma_2 \nabla_\theta \varepsilon_z(h_{\theta^z}, f_z) \; + \frac{1}{1+\mathrm{e}^{-\lambda}}(\theta^{\text{NN}} - \theta^z)$
end for
Update α w.r.t. the second half of $S^{t_{ts}}$: $\alpha = \alpha - \gamma_3 \sum_{\mathcal{T} \sim Pr(\mathcal{T})} \nabla_\alpha \, \varepsilon_{\mathcal{T}}(h_{\theta^{\mathcal{T}}}, f_{\mathcal{T}})$.
end while

of classes. Meta-training is performed via the following episodic train / test splits: two samples are available for each task, the first sample consists of K data points from each class. The additional sample is used both for adaptation and for updating the shared parameters. The algorithm is tested on samples never observed during training.

Omniglot consists of 50 alphabets (tasks) with a total of 1623 handwritten characters (each with 20 examples) [8]. To compare on common ground, we adopt the training procedure defined in [9, 10]. **mini-ImageNet** consists of 60,000 coloured images [9]. It contains 100 classes, with 600 images each. **Multi-dataset OVD** contains a heterogeneous set of tasks [4], from Omniglot, VGG Flower and DTD.

3.1 General FSL Benchmarks

We compare PAML to many seminal ML works on standard FSL benchmarks. The reported results are averages of 600 random trials. The reported standard errors are for 95% confidence intervals. Similar to previous works like [10], we focus on comparisons with methods that do not employ residual networks nor pre-training, to concentrate on assessing the algorithm rather than the power of a deep discriminative model.

Classification results on Omniglot and mini-ImageNet are displayed in Table 1. PAML achieves the joint highest accuracy on 3 out of the 6 K-shot C-way settings and singly achieves new SOTA results on 3 settings (which include all the mini-ImageNet settings). This demonstrates the efficacy of PAML as an ML algorithm on general settings.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Omniglot	5-way accuracy (%)		20-way accuracy (%)		**mIN**	5-way accuracy (%)	
	1-shot	5-shot	1-shot	5-shot		1-shot	5-shot
Matching Nets [9]	98.1	98.9	93.8	98.5		46.6	60.0
MAML [6]	98.7 ± 0.4	$\mathbf{99.9 \pm 0.1}$	95.8 ± 0.3	98.9 ± 0.2		48.7 ± 1.8	63.1 ± 0.9
Prototypical Nets [11]	97.4	99.3	95.4	98.7		46.6 ± 0.8	65.8 ± 0.7
mAP-DLM [12]	98.8	99.6	95.4	98.6		50.3 ± 0.8	63.7 ± 0.7
Relation Net [13]	$\mathbf{99.6 \pm 0.2}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{97.6 \pm 0.2}$	99.1 ± 0.1		50.4 ± 0.8	65.3 ± 0.7
VERSA [10]	$\mathbf{99.7 \pm 0.2}$	$\mathbf{99.8 \pm 0.1}$	$\mathbf{97.7 \pm 0.3}$	98.8 ± 0.2		53.4 ± 1.8	67.4 ± 0.9
Bayesian TAML [4]	94.4 ± 0.6	97.7 ± 0.4	$\mathbf{98.1 \pm 0.3}$	99.1 ± 0.2		51.7 ± 1.6	68.3 ± 1.1
Sharp-MAML [5]	94.7 ± 0.2	95.2 ± 0.3	93.5 ± 0.2	96.6 ± 0.2		50.3 ± 0.7	65.1 ± 0.8
PAML	$\mathbf{99.8 \pm 0.1}$	$\mathbf{99.9 \pm 0.1}$	$\mathbf{98.2 \pm 0.2}$	$\mathbf{99.6 \pm 0.1}$		$\mathbf{58.4 \pm 0.5}$	$\mathbf{72.4 \pm 0.8}$

Table 1: Classification accuracy results for different FSL (K-shot C-way) settings on Omniglot and mini-ImageNet (mIN). A student's t-distribution is used to compute the reported 95% confidence intervals. Bold refers to significance. PAML always achieves SOTA results, either jointly (3 settings) or singly (3 settings).

	10-way any-shot accuracy (%)
MT-NET [14]	94.7 ± 0.6
Bayesian TAML	97.7 ± 0.6
PAML	$\mathbf{99.1 \pm 0.5}$

Table 2: Classification results on OVD which consists of heterogeneous tasks. In these more challenging circumstances, PAML achieves new SOTA results on OVD.

3.2 A Highly Heterogeneous Dataset

The Multi-dataset OVD [4] is a heterogeneous aggregation of three datasets: Omniglot, VGG Flower and DTD. We compare to previous SOTA on the OVD dataset. To enable fair comparison, we adopt an any-shot 10-way setting in this experiment, where the number of shots randomly changes between 1 and 15 for each class.

As displayed in Table 2, PAML achieves a new SOTA result on OVD, with an average classification accuracy improvement of 1.4%. The adaptation mechanism proposed by PAML leads to an ML framework capable of learning in heterogeneous ML environments.

3.3 Other Experimental Details

Values of the step size parameters are: $\gamma_1 = \gamma_2 = \gamma_3 = 0.25$ for Omniglot and OVD, and $\gamma_1 = \gamma_2 = \gamma_3 = 0.01$ for mini-ImageNet. The network architecture is given in Table 3.

Output size	Layers
Shared part (α)	
$14 \times 14 \times 64$	conv2d (3×3, stride 1, SAME, RELU), dropout, pool (2×2, stride 2, SAME)
$7 \times 7 \times 64$	conv2d (3×3, stride 1, SAME, RELU), dropout, pool (2×2, stride 2, SAME)
$4 \times 4 \times 64$	conv2d (3×3, stride 1, SAME, RELU), dropout, pool (2×2, stride 2, SAME)
$2 \times 2 \times 64$	conv2d (3×3, stride 1, SAME, RELU), dropout, pool (2×2, stride 2, SAME)
256	flatten
Task-specific part (θ^t)	
256	$2 \times$ fully connected, ELU + linear, fully connected to $\mu_{\theta^t}, \rho_{\theta^t}$
C (number of classes)	fully-connected softmax

Table 3: Details of the network architecture of PAML.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

4 Conclusion

We introduced a meta-learning algorithm which combines the global meta-learning optimisation with a local adaptation mechanism that addresses each test task via estimating the similarities with previously encountered tasks. We demonstrated strong empirical performance especially on environments comprising highly heterogeneous tasks.

References

[1] B. Lake, T. Ullman, J. Tenenbaum, and S. Greshman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[2] J. Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. *PhD thesis, TUM*, 1987.

[3] R. Vuorio, S. Sun, H. Hu, and J. Lim. Multimodal model-agnostic meta-Learning via task-aware modulation. *NeurIPS*, 2019.

[4] D. Na, B. Lee, H. Lee, S. Kim, M. Park, E. Yang, and S. Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. *ICLR*, 2020.

[5] M. Abbas, Q. Xiao, L. Chen, P. Chen, and T. Chen. Sharp-MAML: Sharpness-aware model-agnostic meta learning. *ICML*, 2022.

[6] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 34, 2017.

[7] Y. Li and P. Long. Learnability and the doubling dimension. *NeurIPS*, 2007.

[8] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum. One shot learning of simple visual concepts. *Proceedings of the Cognitive Science Society*, 33, 2011.

[9] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. *NeurIPS*, 2016.

[10] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. Turner. Meta-learning probabilistic inference for prediction. *ICLR*, 2019.

[11] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

[12] E. Triantafillou, R. Zemel, and R. Urtasun. Few-shot learning through an information retrieval lens. *NeurIPS*, 2017.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. Torr, and T. Hospedales. Learning to compare: Relation network for few-shot learning. *CVPR*, 2018.

[14] Y. Lee and S. Choi. Gradient-based meta-learning with learned layerwise metric and subspace. *ICML*, 2018.