LEARNING WHERE TO LEARN

Dominic Zhao, Nicolas Zucchet, João Sacramento[*], Johannes von Oswald[*]
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{dozhao, nzucchet, rjoao, voswaldj}@ethz.ch

ABSTRACT

Finding neural network weights that generalize well from small datasets is difficult. A promising approach is to (meta-)learn a weight initialization from a collection of tasks, such that a small number of weight changes results in low generalization error. We show that this form of meta-learning can be improved by letting the learning algorithm decide which weights to change, i.e., by learning where to learn. We find that patterned sparsity emerges from this process. Lower-level features tend to be frozen, while weights close to the output remain plastic. This selective sparsity enables running longer sequences of weight updates without overfitting, resulting in better generalization in the miniImageNet benchmark. Our findings shed light on an ongoing debate on whether meta-learning can discover adaptable features, and suggest that sparse learning can outperform simpler feature reuse schemes.

1 INTRODUCTION

The well-known model-agnostic meta-learning (MAML; Finn et al., 2017) algorithm aims to learn a neural network initialization such that adaptations to new tasks generalize well. This strategy has proven useful in the few-shot learning setting, where training sets are small and overfitting can easily occur. In MAML, this problem is partially mitigated by limiting model adaptation to only a few gradient steps, which can be seen as an 'early stopping'. Here, we shed light on the inner workings of MAML and advocate a *regularization by sparse learning* approach. Our results question the benefits of more sophisticated meta-learners that modulate gradients (Li et al., 2017; Zintgraf et al., 2019; Flennerhag et al., 2020; Lee & Choi, 2018; Zhao et al., 2020; Chen et al., 2020).

Our study builds upon the surprising effectiveness of almost no inner-loop training (ANIL; Raghu et al., 2020). It has been recently shown that applying MAML while adapting only last-layer weights leads to almost no decrease in performance in standard few-shot learning benchmarks. Instead of deciding which weights to freeze a priori, here we endow the meta-learner with the possibility to explicitly stop optimizing certain weights in the inner-loop learning process. We do this by introducing an adjustable binary mask which is elementwise multiplied with gradient updates. Overfitting can thus be prevented and learning sped-up by focusing adaptation to a sparse parameter subset, discovered by MAML.

We find that our sparse-MAML algorithm recovers similar behavior to ANIL. It induces high update sparsity in earlier layers of the network while allowing for adaptation in deeper layers including the network's output. These findings are also in line with accumulating evidence for reduced plasticity in lower-order sensory areas of the adult brain, after an initial developmental phase of high plasticity (Wandell & Smirnakis, 2009; Lohmann & Kessels, 2014). Interestingly, sparsity adapts intuitively to the number of inner loop gradient steps as well as its learning rate, the few-shot dataset size and network specifications. This leads to a simple, robust and interpretable variant of MAML that improves generalization by self-regularizing the parameters that the model should learn.

2 FROM MAML TO SPARSE-MAML

Few-shot learning aims to find a network that performs well when trained on few samples of unseen data. Formally, consider a set of small tasks $\{\tau_i\}$, with each task τ_i containing a training $\mathcal{D}_i^{\text{t}}$

[*] equal contribution

and validation $\mathcal{D}_i^{\text{v}}$ dataset. A loss, $\mathcal{L}(\phi; \mathcal{D})$, measures the quality of the prediction of a network parametrized by ϕ on the dataset $\mathcal{D}$. Few shot learning then consists in optimizing the parameters θ of a learning algorithm $\mathcal{A}$ that given the training dataset produces the parameters ϕ in order to improve performance on the validation set, i.e.,

$$\min_{\theta} \mathbb{E}_i \big[\mathcal{L}(\mathcal{A}(\theta; \mathcal{D}_i^{\text{t}}), \mathcal{D}_i^{\text{v}})\big].$$

MAML In MAML, this problem is approached by looking for network weights from which few gradient descent steps are needed to reach high performance. The resulting optimization problem can be formulated as follows:

$$\min_{\theta} \mathbb{E}_i \big[\mathcal{L}\left(\phi_K(\theta, \mathcal{D}_i^{\text{t}}), \mathcal{D}_i^{\text{v}}\right)\big] \text{ s.t. } \phi_{k+1} = \phi_k - \alpha \nabla_\phi \mathcal{L}\left(\phi_k, \mathcal{D}_i^{\text{t}}\right) \text{ and } \phi_0 = \theta,$$

with α the inner loop learning rate and K the number of gradient descent steps. The initialization θ is then obtained by iterative updating, using

$$\theta \leftarrow \theta - \gamma \, \mathrm{d}_\theta \, \mathbb{E}_i \big[\mathcal{L}\left(\phi_K(\theta, \mathcal{D}_i^{\text{t}}), \mathcal{D}_i^{\text{v}}\right)\big],$$

with γ the outer loop learning rate. The derivative w.r.t. to θ requires backpropagating through the inner optimization and is thus resource-intensive. First-order MAML (FOMAML) drastically reduces the cost by setting to zero the second-order derivatives that appear when differentiating the inner loop update.

Learning the learning rates Some variants of MAML focus on learning the learning rate. More precisely, we consider inner loop updates of the following form:

$$\phi_{k+1} = \phi_k - \alpha M \nabla_\phi \mathcal{L}\left(\phi_k, \mathcal{D}_i^{\text{t}}\right),$$

for some learnable preconditioning matrix M. Through M, we learn some information on the geometry of the loss with the hope of faster inner loop optimization. It is updated similarly to θ. Meta-SGD (Li et al., 2017) considers a diagonal M, i.e. learnable learning rates, and Meta-Curvature (Park & Oliva, 2019) considers a block matrix. Note that MAML corresponds to the $M = \text{Id}$ case.

Sparse-MAML Following those approaches, we introduce sparse-MAML. It dynamically learns the parameters which will be updated and the ones that won't. Hence, sparse-MAML learns where to learn. To do so, we use a vector m (instead of a matrix M) that modulates the gradient in the inner loop update the following way:

$$\phi_{k+1} = \phi_k - \alpha\big(\mathbb{1}_{m \geq 0} \circ \nabla_\phi \mathcal{L}\left(\phi_k, \mathcal{D}_i^{\text{t}}\right)\big),$$

with $\mathbb{1}_{\cdot \geq 0} : \mathbb{R}^n \to \{0, 1\}^n$ the step function that we apply elementwise to the underlying parameter vector $m \in \mathbb{R}^n$ and $\circ$ the pointwise multiplication. We differentiate the step function by considering it linear; a method called the straight-through estimator (Bengio et al., 2013) that was recently used for similar large-scale masking (Ramanujan et al., 2020). Following FOMAML, we ignore second-order derivatives. This leads to the update

$$m \leftarrow m + \alpha \, \gamma \, \mathbb{E}_i \left[\nabla_\phi \mathcal{L}(\phi_K, \mathcal{D}_i^{\text{v}}) \circ \sum_{k=0}^{K-1} \nabla_\phi \mathcal{L}\big(\phi_k, \mathcal{D}_i^{\text{t}}\big)\right].$$

A detailed derivation can be found in Appendix B. Our mask update depends on the alignment between the validation loss gradient g_i^{v} and the training loss gradient $\overline{g}_i^{\text{t}} := \sum_{k=0}^{K-1} \nabla_\phi \mathcal{L}(\phi_k, \mathcal{D}_i^{\text{t}})$ averaged over the inner loop trajectory. Learning tends to shut off for coordinates for which these two quantities are of opposing sign, $\mathbb{E}_i[g_i^{\text{v}} \, \overline{g}_i^{\text{t}}] < 0$. This freezing of learning when the parameter updates are conflicting on the training and validation sets can be beneficial for generalization.

3 Experiments

Our main aim is to investigate if and how sparse-MAML uses the possibility to prevent updating weights for different training regimes. We focus on two hyperparameters, the inner loop learning rate and the number of inner loop gradient steps. We define the update sparsity of a parameter group or the entire network as $1 - \|\mathbb{1}_{m \geq 0}\|^2 / \dim(m)$ and use it to monitor the state of the mask. Unless explicitly stated, we follow the experimental setup of (Finn et al., 2017; Vinyals et al., 2016) and focus on the common non-saturated benchmark MiniImagenet (Ravi & Larochelle, 2016) for two data regimes: 5-shot 5-way and 1-shot 5-way. All experimental details can be found in Appendix A.

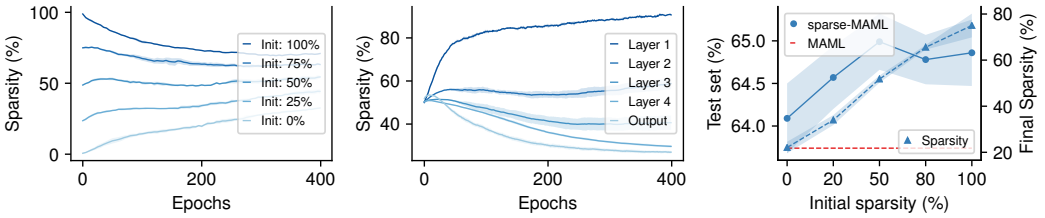

Figure 1: Weight update sparsity emerges in 5-shot 5-way classification of MiniImagenet (Ravi & Larochelle, 2016). Results averaged over 5 seeds $\pm$ std. *Left*: Averaged network sparsity adapts for different sparsity initializations throughout training. *Center*: Different final update sparsity for convolutional weight matrices and the network's output layer emerge with gradually less sparsity from earlier to deeper layers while all being initialized at 50% sparsity. *Right:* Sparse-MAML shows improved test set accuracy for higher update sparsity initializations.

3.1 Layer specific sparsity

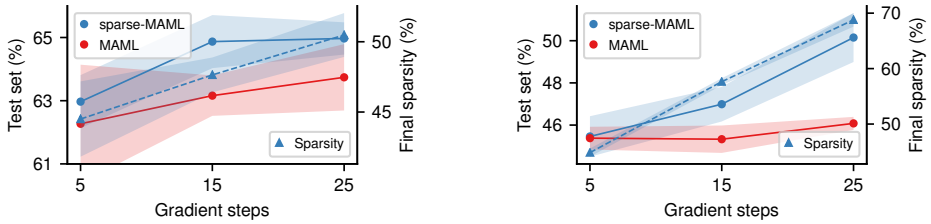

Figure 2: Update sparsity adjusts to the number of gradient steps in 5-shot 5-way (*left*) and 1-shot 5-way (*right*) classification of MiniImagenet (Ravi & Larochelle, 2016). In both data regimes, more steps in the inner loop lead to higher final update sparsity and better generalization compared to MAML. Note that in the lower data regime (*right*) much higher update sparsity emerges although the exact same model and training setup is used. In all experiments the update sparsity is initialized at 50%, the inner loop learning rate is 0.1 and results are averaged over 5 seeds $\pm$ std.

Our first finding validates and extends the phenomenon described in Raghu et al. (2020). As shown in Figure 1, sparse-MAML dynamically adjusts the update sparsity of the network, with very different values over the layers. As an example, we show the average update sparsity of the 4 convolutional weight matrices and the output layer during training. The same trend is observed for other parameter groups in the network except the output bias (see Figure 4 in Appendix A). Sparsity clearly correlates with the depth of the network parameter and gradually increases towards the early layers of the network, despite the similar value before training (around 50%). This refines the findings of Raghu et al. (2020) by showing that sparse-MAML suppresses inner loop updates of weights in earlier layers while allowing deeper layers to adjust to new tasks. This dynamic sparsity adjustment is robust across different sparsity initializations (Figure 1, left plot) while increasing few-shot test set accuracy robustly over all sparsity initializations (Figure 1, right plot).

3.2 Sparsity adjusts to hyperparameters

We study the effects of inner loop learning rate and length on the final update sparsity. First, we test three different inner loop durations (5, 15 or 25 gradient steps, see Figure 2). We find that neither MAML nor sparse-MAML exhibit overfitting for the duration range considered here (for reference, the original study of MAML applied 5 inner-loop steps during meta-training). On the contrary, the solutions found by sparse-MAML generalize significantly better for *longer* adaptation phases. This improvement in generalization is accompanied by an increase in update sparsity.

To further investigate if increasing model adaptability to new tasks can result in improved generalization, in combination with update sparsity, we simply scan the inner loop learning rate over a large range, see left plot of Figure 3. Here, we find a clear trend towards higher network sparsity in hand with better test set accuracy for larger learning rates. Note that deep layers remain highly plastic (not shown). Interestingly, similar effects have been reported in classic neural network training where

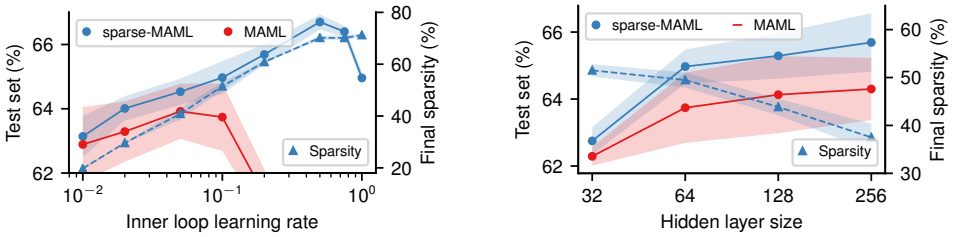

Figure 3: Update sparsity for different hyperparameters for 5-shot 5-way classification of MiniImagenet (Ravi & Larochelle, 2016). Figures show results for a sparsity initialization at 50% averaged over 5 seeds $\pm$ std. *Left*: Higher inner loop learning rates enforce higher sparsity with gradually better test set generalization. MAML fails to adjust to high learning rates. *Right:* Update sparsity adjusts to different hidden layer sizes of the 4-layer network with less sparsity for wider networks. The inner loop learning rate was set 0.1 for all hidden sizes.

both freezing layers throughout training (Raghu et al., 2017; Brock et al., 2017) and the use of large learning rates (Lewkowycz et al., 2020) seem to improve generalization.

Finally, we also find that sparsity correlates strongly with the width of the network, decreasing for networks with larger hidden size, see right plot of Figure 3. We leave a more thorough investigation of sparse-MAML on wider and deeper networks as well as the use of meta-learning algorithms that intrinsically use long inner loops (Rajeswaran et al., 2019; Zucchet et al., 2021) for future research.

3.3 Sparse-MAML and Meta-SGD

While strictly less expressive, sparse-MAML is arguably conceptually simpler and more interpretable than Meta-SGD. This allows carrying out the analyses presented in the previous sections. Despite its simplicity, sparse-MAML performs on par with Meta-SGD on the two investigated few-shot regimes, cf. Table 1. Meta-SGD and Meta-Curvature (MC) take a single gradient step in the inner loop (with a small learning rate of 0.001), which prevents overfitting. We further note that the results for Meta-Curvature are not directly comparable as additional data augmentation was used. We adjourn investigating fusions of Meta-SGD and mechanisms to stop learning using gradient modulation like $\text{ReLU}(m) \circ \nabla_\phi \mathcal{L}(\phi_k; \mathcal{D}_i^t)$.

Table 1: 5-way Few-shot classification accuracy (%) on MiniImagenet. Mean $\pm$ std. over 5 seeds.

Method	1-shot (↑)	5-shot (↑)
MAML	$48.07^{\pm 1.75}$	$63.15^{\pm 0.91}$
ANIL	$46.70^{\pm 0.40}$	$61.50^{\pm 0.50}$
Meta-SGD	$50.47^{\pm 1.87}$	$64.03^{\pm 0.94}$
MC (+data aug.)	$54.23^{\pm 0.88}$	$68.47^{\pm 0.69}$
sparse-MAML	$50.15^{\pm 1.19}$	$66.70^{\pm 0.23}$

4 Conclusion

We investigate a simple variant of MAML termed sparse-MAML, a meta-learner that is capable of learning where to learn. This enables systematically analyzing update sparsity as a function of network depth, shedding light into the inner workings of MAML on few-shot classification problems. A clear trend emerges: the model should remain flexible close to the output, and learning should essentially stop in the first layers. The meta-learning process therefore discovers a finer version of the recent ANIL algorithm, whose inner loop consists in adapting only the last-layer weights. Moreover, the level of sparsity adaptively changes with model architecture, data-set size and inner-loop optimization hyperparameters, leading to robust improved generalization. We hope that our analyses further stimulates the debate on optimization-based meta-learners that offer a powerful and general framework for learning-to-learn.

Acknowledgements

This work was supported by an Ambizione grant (PZ00P3186027) awarded to J.S. from the Swiss National Science Foundation. Johannes von Oswald is funded by the Swiss Data Science Center (J.v.O. P18-03). Dominic Zhao is supported by AlayaLabs (Montreal, Canada) . We thank Frederik

Benzing, Angelika Steger, Laura Sainz, Massimo Caccia, Seijin Kobayashi and Simon Schug for helpful comments.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. FreezeOut: Accelerate training by progressively freezing layers. *arXiv preprint arXiv:1706.04983*, 2017.

Yutian Chen, Abram L. Friesen, Feryal Behbahani, David Budden, Matthew W. Hoffman, Arnaud Doucet, and Nando de Freitas. Modular meta-learning with shrinkage. In *Advances in Neural Information Processing Systems*, 2020.

Tristan Deleu, Tobias Würfl, Mandana Samiei, Joseph Paul Cohen, and Yoshua Bengio. Torchmeta: A meta-learning library for PyTorch. *arXiv preprint arXiv:1909.06576*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Sebastian Flennerhag, Andrei A. Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. In *International Conference on Learning Representations*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision*, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, 2018.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-SGD: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Christian Lohmann and Helmut W Kessels. The developmental stages of synaptic plasticity. *The Journal of Physiology*, 592(1):13–31, 2014.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Eunbyung Park and Junier B. Oliva. Meta-curvature. In *Advances in Neural Information Processing Systems*, 2019.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. In *International Conference on Learning Representations*, 2020.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for Deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, 2017.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-Learning with Implicit Gradients. In *Advances in Neural Information Processing Systems*, 2019.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

Brian A Wandell and Stelios M Smirnakis. Plasticity and stability of visual field maps in adult primary visual cortex. *Nature Reviews Neuroscience*, 10(12):873–884, 2009.

Dominic Zhao, Johannes von Oswald, Seijin Kobayashi, João Sacramento, and Benjamin F Grewe. Meta-learning via hypernetworks. In *NeurIPS Workshop on Meta-Learning*, 2020.

Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, 2019.

Nicolas Zucchet, Simon Schug, Johannes von Oswald, Dominic Zhao, and João Sacramento. A contrastive rule for meta-learning. *arXiv preprint arXiv:2104.01677*, 2021.

A Experimental Setups

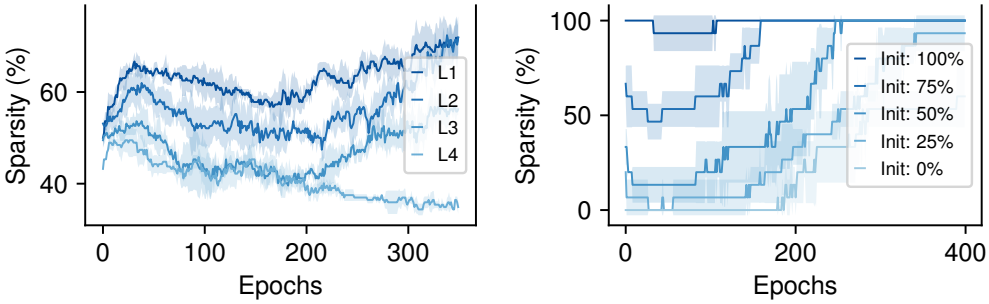

Figure 4: Emergent weight update sparsity in 5-shot 5-way classification of MiniImagenet (Ravi & Larochelle, 2016) with a 4-layer convolutional neural network, inner loop learning rate 0.1 and 25 inner loop steps. Results averaged over 5 seeds $\pm$ std. *Left*: Different final update sparsity for BatchNorm gain parameters emerge with gradually less sparsity from earlier to deeper layers while all being initialised at 50% sparsity. *Right:* Output layer bias parameter sparsity for different sparsity initialisation tend towards 100%. Note that usually deeper layers tend towards lower sparsity.

All experiments, unless specified otherwise, follow the few-shot learning classification experimental setup proposed in (Finn et al., 2017) and are performed on the MiniImageNet dataset introduced by (Ravi & Larochelle, 2016; Vinyals et al., 2016) which consists of 64 training classes, 12 validation classes and 24 test classes. The backbone classifier consists of four convolutional layers each with 64-filters followed by a BatchNorm layer (Ioffe & Szegedy) as well as max-pooling layer with kernel size and stride of 2. The network then projects to its output via a fully-connected layer.

When not being the subject of analysis, the hyperparameters for the classification results are the following. For 1-shot and 5-shot experiments, we use batches of sizes 4 resp. 2, with 25 gradient descent inner-loop steps during meta-training and meta-testing. When not stated otherwise an inner learning rate 0.1 is used. For the outer-loop optimisation we use ADAM (Kingma & Ba (2015)) with a learning rate of 0.001 and PyTorch default parameter for both the meta-parameters as well as parameters underlying the gradient masks. We train all models for 400 epochs of 100 training tasks each. All weight matrices including m, the gradient mask parameters, are initialised with Kaiming initialisation (He et al., 2015).

The reported test set accuracies including the sparsity values are based on models that are checkpointed on the accuracies averaged across 500 tasks build from the validation set. The model with

best average validation set accuracy is then tested on 300 tasks of the test set data. We note that all experiments, dataset split as well as meta-gradient computations used the Torchmeta library version 1.6 (Deleu et al., 2019).

We stress that we use first-order MAML (Finn et al., 2017) to learn ϕ_0 in all of the reported results and handle the BatchNorm parameters as in the *transductive* learning protocol as originally done in MAML (Finn et al., 2017; Nichol et al., 2018).

Results in Table 1 are the best values found by our scans of learning rates and inner loop steps using a sparsity initialization of 50%. For the 1-shot and the 5-shot regime, 35 steps and an inner learning rate of 0.5 produce the reported test set accuracy.

B Derivation of the sparse-MAML update

We here derive the sparse-MAML update rule on the mask, that is

$$m \leftarrow m + \alpha\,\gamma\,\mathbb{E}_i\left[\nabla_\phi \mathcal{L}(\phi_K, \mathcal{D}_i^{\text{v}}) \circ \sum_{k=0}^{K-1} \nabla_\phi \mathcal{L}\big(\phi_k, \mathcal{D}_i^{\text{t}}\big)\right].$$

For the sake of clarity, we omit the dependencies of ϕ_K on θ and $\mathcal{D}_i^{\text{v}}$ in the following. We start by exchanging the derivative and expectation

$$\mathrm{d}_m \mathbb{E}_i[\mathcal{L}(\phi_K, \mathcal{D}_i^{\text{v}})] = \mathbb{E}_i[\mathrm{d}_m \mathcal{L}(\phi_K, \mathcal{D}_i^{\text{v}})]$$

and applying the chain rule

$$\mathrm{d}_m \mathcal{L}\left(\phi_K, \mathcal{D}_i^{\text{v}}\right) = \nabla_\phi \mathcal{L}\left(\phi_K, \mathcal{D}_i^{\text{v}}\right) \mathrm{d}_m \phi_K.$$

We now compute the derivative of ϕ_K with respect to the mask:

$$\mathrm{d}_m \phi_K = \mathrm{d}_m \phi_{K-1} - \alpha\, \mathrm{d}_m \big[\mathbb{1}_{m\geq 0} \circ \nabla_\phi \mathcal{L}\left(\phi_{K-1}, \mathcal{D}_i^{\text{t}}\right)\big].$$

To avoid computing second-order derivatives and following FOMAML, we consider $\nabla_\phi \mathcal{L}\left(\phi_{K-1}, \mathcal{D}_i^{\text{t}}\right)$ to be constant with respect to m. It remains

$$\mathrm{d}_m \phi_K \approx \mathrm{d}_m \phi_{K-1} - \alpha\, \mathrm{d}_m \left[\mathbb{1}_{m\geq 0}\right] \cdot \mathrm{diag}\big(\nabla_\phi \mathcal{L}\left(\phi_{K-1}, \mathcal{D}_i^{\text{t}}\right)\big).$$

We then approximate $\mathrm{d}_m \mathbb{1}_{m\geq 0}$ using straight-through estimation, which consists in taking the derivative equal to the identity. We thus have

$$\mathrm{d}_m \phi_K \approx \mathrm{d}_m \phi_{K-1} - \alpha\, \mathrm{diag}\big(\nabla_\phi \mathcal{L}\big(\phi_{K-1}, \mathcal{D}_i^{\text{t}}\big)\big)$$

and

$$\mathrm{d}_m \phi_K \approx -\alpha \sum_{k=0}^{K-1} \mathrm{diag}\big(\nabla_\phi \mathcal{L}\left(\phi_k, \mathcal{D}_i^{\text{t}}\right)\big).$$

Finally, we use the meta-gradient approximation

$$\mathrm{d}_m \mathbb{E}_i\big[\mathcal{L}\big(\phi_K(\theta, \mathcal{D}_i^{\text{t}}), \mathcal{D}_i^{\text{v}}\big)\big] \approx -\alpha\, \mathbb{E}_i\left[\nabla_\phi \mathcal{L}\left(\phi_K, \mathcal{D}_i^{\text{v}}\right) \circ \sum_{k=0}^{K-1} \nabla_\phi \mathcal{L}\big(\phi_k, \mathcal{D}_i^{\text{t}}\big)\right]$$

and m is updated with

$$m \leftarrow m + \alpha\,\gamma\,\mathbb{E}_i\left[\nabla_\phi \mathcal{L}(\phi_K, \mathcal{D}_i^{\text{v}}) \circ \sum_{k=0}^{K-1} \nabla_\phi \mathcal{L}\big(\phi_k, \mathcal{D}_i^{\text{t}}\big)\right].$$