RECASTING GRADIENT-BASED META-LEARNING AS HIERARCHICAL BAYES

Anonymous authors
Paper under double-blind review

ABSTRACT

Meta-learning allows an intelligent agent to leverage prior learning episodes as a basis for quickly improving performance on a novel task. Bayesian hierarchical modeling provides a theoretical framework for formalizing meta-learning as inference for a set of parameters that are shared across tasks. We reformulate the model-agnostic meta-learning algorithm (MAML) by Finn et al. (2017) as a method for probabilistic inference in a hierarchical Bayesian model. In contrast to prior methods for meta-learning via hierarchical Bayes, MAML is naturally applicable to complex function approximators through its use of a scalable gradient descent procedure for posterior inference. Furthermore, the identification of MAML as probabilistic inference provides a way to understand the algorithm's operation as a meta-learning procedure, as well as an opportunity to make use of computational strategies for Bayesian inference. We use this opportunity to propose an improvement to the MAML algorithm inspired by approximate Bayesian posterior inference, and show increased performance on a few-shot learning benchmark.

1 INTRODUCTION

A remarkable aspect of human intelligence is the ability to quickly solve a novel problem, and to do so even in the face of limited experience in a novel domain. Such fast adaptation is made possible by leveraging prior learning experience in order to improve the efficiency of later learning. This capacity for *meta-learning* also has the potential to enable an artificially intelligent agent to learn more efficiently in situations with little available data or limited computational resources (Schmidhuber, 1987; Bengio et al., 1991; Naik & Mammone, 1992).

In machine learning and reinforcement learning, the problem of meta-learning has been formulated as the minimization of an aggregate loss over a family of related tasks (Caruana, 1993; Thrun & Pratt, 1998). Previous work has demonstrated improved performance on meta-learning benchmarks by employing models that encode the structure of the meta-learning problem in the architecture or loss function (*e.g.,* Vinyals et al., 2016), or by using a recurrent neural network (RNN) as a form of memory augmentation to encode and retrieve episodic information (*e.g.,* Santoro et al., 2016).

In contrast to architectural or RNN-based meta-learning methods, model-agnostic meta-learning (MAML) (Finn et al., 2017) provides a gradient-based meta-learning procedure that directly uses the standard gradient descent learning rule to adapt quickly to each novel task. Fast adaptation is made possible by learning a set of initial parameters that is shared among the task-specific models. The intuition behind this approach is that gradient descent from the learned initialization provides a favorable inductive bias for fast adaptation. However, this inductive bias is evaluated only empirically in prior work (Finn et al., 2017).

In this work, we present a novel derivation of and a novel extension to MAML, illustrating that this algorithm can be understood as inference for the parameters of a prior distribution in a hierarchical Bayesian model. The learned prior allows for quick adaptation to unseen tasks on the basis of an implicit predictive density over task-specific parameters. The reinterpretation as hierarchical Bayes gives a principled statistical motivation for MAML as a meta-learning algorithm, and sheds light on the potential reasons for its favorable performance even amongst methods with significantly more parameters. More importantly, by casting gradient-based meta-learning within a Bayesian framework, we are able to improve MAML by taking insights from Bayesian posterior estimation as novel augmentations to the gradient-based meta-learning procedure. We experimentally demonstrate that this enables better performance on a few-shot learning benchmark.

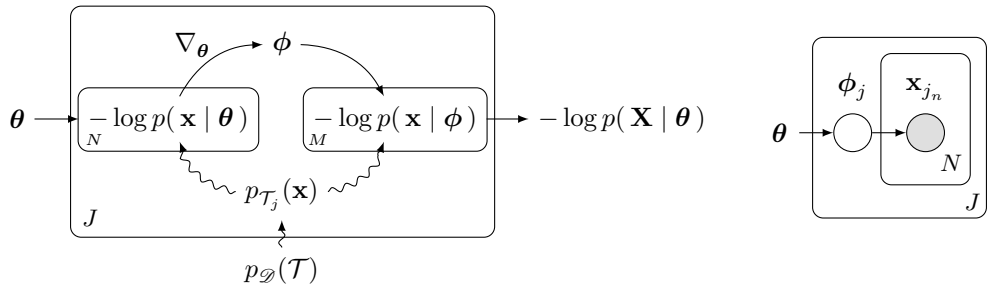


Figure 1: **(Left)** The computational graph of the MAML (Finn et al., 2017) algorithm covered in Section 2.1. **(Right)** The probabilistic graphical model for which MAML provides an inference procedure as described in Section 3.1. Plates denote repeated computations (**left**) or factorization (**right**) across independent and identically distributed samples.

2 Meta-Learning Formulation

The goal of a meta-learner is to extract task-general knowledge through the experience of solving a number of related tasks. By using this learned prior knowledge, the learner has the potential to quickly adapt to novel tasks even in the face of limited data or limited computation time.

Formally, we consider a dataset $\mathscr{D}$ that defines a distribution over family of tasks $\mathcal{T}$. These tasks share some common structure such that learning to solve a single task has the potential to aid in solving another. Each task $\mathcal{T}$ defines a distribution over datapoints $\mathbf{x}$, which we assume in this work to consist of inputs and either regression targets or classification labels in a supervised learning problem (although this assumption can be relaxed; *e.g.,* see Finn et al., 2017). The objective of the meta-learner is to optimize a task-specific performance metric associated with any given unseen task from the dataset, *i.e.,* to be capable of fast adaptation to a novel task.

In the following subsections, we discuss two ways of formulating a solution to the meta-learning problem. These approaches are orthgonal, but, as we show in Section 3.1, a specific gradient descent procedure, MAML, provides a bridge between the two.

2.1 Meta-Learning as Gradient-Based Optimization

A gradient-based meta-learner aims to find some shared parameters $\boldsymbol{\theta}$ that make it easier to find the right task-specific parameters $\boldsymbol{\phi}$ when faced with a novel task. A variety of meta-learners that employ gradient methods for fast adaptation have been proposed (*e.g.,* Li & Malik, 2017a; Andrychowicz et al., 2016; Li & Malik, 2017b; Wichrowska et al., 2017). MAML (Finn et al., 2017) is distinct in that it provides a gradient-based meta-learning procedure that employs no additional paramaters and operates on the same weights for both meta-learning and task-specific adaptation. These are necessary features for the equivalence we show in Section 3.1.

To solve the meta-learning problem, MAML estimates the parameters $\boldsymbol{\theta}$ of a set of models so that when one or a few batch gradient descent steps are taken from the initialization at $\boldsymbol{\theta}$ given a small sample of data $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_{\mathcal{T}}(\mathbf{x})$ each model has good generalization performance on another sample $\mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \sim p_{\mathcal{T}}(\mathbf{x})$. The MAML objective in a maximum likelihood setting is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{J} \sum_{j} \big[-\log p\big(\mathbf{x}_{j_{N+1}}, \ldots, \mathbf{x}_{j_{N+M}} \mid \underbrace{\boldsymbol{\theta} + \alpha \, \nabla_{\boldsymbol{\theta}} \log p\left(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\theta} \right)}_{\boldsymbol{\phi}_j} \big) \big] \tag{1}$$

where we use $\boldsymbol{\phi}_j$ to denote the updated parameters after taking a single batch gradient descent step with step size α on the negative log likelihood associated with the task $\mathcal{T}_j$. We refer to the inner gradient descent procedure that computes $\boldsymbol{\phi}_j$ as *fast adaptation*. The computational graph for MAML is given in Figure 1 (left).

2.2 Meta-Learning as Probabilistic Inference

An alternative way to formulate the meta-learning problem is as a problem of probabilistic inference in the hierarchical model depicted in Figure 1 (right). In particular, in the case of meta-learning, each task-specific parameter $\boldsymbol{\phi}$ is distinct from but should share statistical strength with the parameters

from other tasks. We can capture this intuition by introducing a meta-level parameter $\boldsymbol{\theta}$ on which the task-specific parameters are statistically dependent.

Formally, grouping all data as $\mathbf{X}$ and denoting by $\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N}$ a sample from task $\mathcal{T}_j$, the marginal likelihood of the observed data is given by

$$p\left(\mathbf{X} \mid \boldsymbol{\theta}\right) = \prod_j \left(\int p\left(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\phi}_j\right) p\left(\boldsymbol{\phi}_j \mid \boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\phi}_j \right) \quad . \tag{2}$$

Maximizing (2) as a function of $\boldsymbol{\theta}$ gives a point estimate for $\boldsymbol{\theta}$, a method known as empirical Bayes (Bernardo & Smith, 2006; Gelman et al., 2014) due to its use of the data to estimate the prior parameters.

Hierarchical Bayesian models have a long history of use in both meta-learning and domain adaptation problems (*e.g.,* Lawrence & Platt, 2004; Yu et al., 2005; Gao et al., 2008; Daumé III, 2009; Wan et al., 2012). However, the formulation of meta-learning as hierarchical Bayes does not automatically provide an inference procedure, and furthermore it is not straightforward to integrate such probabilistic models with flexible and expressive deep networks. Moreover, since the task-specific parameters are treated as hidden variables, direct computation of the marginal likelihood in (2) requires an expensive marginalization over $\boldsymbol{\phi}_j$ unless an approximate inference procedure were provided.

3 Connecting Gradient-Based Meta-Learning and Probabilistic Inference

In this section, we connect the two independent approaches of Section 2.1 and Section 2.2 by showing that MAML can be understood as empirical Bayes in a hierarchical probabilistic model. Furthermore, we build on this understanding by showing that a choice of update rule for the meta-level parameters $\boldsymbol{\theta}$ (*i.e.,* a choice of meta-optimizer) corresponds to a choice of prior over task-specific parameters.

3.1 Model-Agnostic Meta-Learning as Empirical Bayes

In general, when performing empirical Bayes, the marginalization over task-specific parameters $\boldsymbol{\phi}$ in (2) is not tractable to compute exactly. To skirt this issue, we can consider approximations that makes use of a point estimate $\hat{\boldsymbol{\phi}}$ instead of performing the integration in (2). Using $\hat{\boldsymbol{\phi}}_j$ as an estimator for each $\boldsymbol{\phi}_j$, we may write the negative logarithm of the marginal likelihood as

$$-\log p\left(\mathbf{X} \mid \boldsymbol{\theta}\right) \approx \sum_j \left[-\log p\left(\mathbf{x}_{j_{N+1}}, \ldots \mathbf{x}_{j_{N+M}} \mid \hat{\boldsymbol{\phi}}_j\right)\right] \quad . \tag{3}$$

Setting $\hat{\boldsymbol{\phi}}_j = \boldsymbol{\theta} + \beta \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\theta})$ for each j in (3) recovers an unscaled version of the one-step MAML objective in (1). This tells us that the MAML objective is equivalent to a maximization with respect to the meta-level parameters $\boldsymbol{\theta}$ of the marginal likelihood, where a point estimate for each task-specific parameter $\boldsymbol{\phi}_j$ is computed via one or a few steps of gradient descent.

We can better understand the point estimate computed by gradient descent by considering the linear regression case. Recall that the *maximum a posteriori* (MAP) estimate of $\boldsymbol{\phi}$ corresponds to the global mode of the posterior $p(\boldsymbol{\phi}_j \mid \mathbf{x}_{j_1}, \ldots \mathbf{x}_{j_N}, \boldsymbol{\theta}) \propto p(\mathbf{x}_{j_1}, \ldots \mathbf{x}_{j_N} \mid \boldsymbol{\phi}_j) p(\boldsymbol{\phi}_j \mid \boldsymbol{\theta})$. In the case of a linear model, it can be shown that early stopping of a regularized iterative optimization procedure to estimate $\boldsymbol{\phi}$ is exactly equivalent to MAP estimation of $\boldsymbol{\phi}$ under the assumption of a prior that depends on the form of the regularization applied. In particular, writing the design matrix as $\mathbf{X}$ and the vector of regression targets as $\mathbf{y}$, consider the gradient descent update

$$\boldsymbol{\phi}_{(k)} = \boldsymbol{\phi}_{(k-1)} - \beta \nabla_{\boldsymbol{\phi}_{(k-1)}} \left[\|\mathbf{y} - \mathbf{X}\boldsymbol{\phi}\|_2^2 \right] = \boldsymbol{\phi}_{(k-1)} - \beta \mathbf{X}^{\mathrm{T}} \left(\mathbf{X}\boldsymbol{\phi}_{(k-1)} - \mathbf{y} \right) \tag{4}$$

for iteration index k and learning rate $\beta \in \mathbb{R}^+$. Santos (1996) shows that, starting from $\boldsymbol{\phi}_{(0)} = \boldsymbol{\theta}$, $\boldsymbol{\phi}_{(k)}$ in (4) solves the regularized linear least squares problem

$$\min \left(\|\mathbf{y} - \mathbf{X}\boldsymbol{\phi}\|_2^2 + \|\boldsymbol{\theta} - \boldsymbol{\phi}\|_{\mathbf{Q}}^2 \right) \tag{5}$$

Algorithm MAML-HB ($\mathscr{D}$)

Initialize $\boldsymbol{\theta}$ randomly
while *not converged* **do**
  Draw J samples $\mathcal{T}_1, \ldots, \mathcal{T}_J \sim p_{\mathscr{D}}(\mathcal{T})$
  Construct marginal NLL estimates $f_{\boldsymbol{\theta}}(\mathcal{T}_1), \ldots, f_{\boldsymbol{\theta}}(\mathcal{T}_J)$ using ML-$\cdots$
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \sum_j f_{\boldsymbol{\theta}}(\mathcal{T}_j)$
end

Algorithm 2: Model-agnostic meta-learning as hierarchical Bayesian inference. The choices of the subroutine ML-$\cdots$ are defined in Subroutine 3 and Subroutine 4.

Subroutine ML-POINT ($\boldsymbol{\theta}, \mathcal{T}$)

Draw N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_{\mathcal{T}}(\mathbf{x})$
Initialize $\boldsymbol{\phi} \leftarrow \boldsymbol{\theta}$
for k *in* $1, \ldots, K$ **do**
  Update $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \beta \nabla_{\boldsymbol{\phi}} \log p(\, \mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\phi}\,)$
end
Draw M samples $\mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \sim p_{\mathcal{T}}(\mathbf{x})$
return $-\log p(\, \mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \mid \boldsymbol{\phi}\,)$

Subroutine 3: Subroutine for computing a point estimate of the marginal negative log likelihood (NLL) using truncated gradient descent.

with $\mathbf{Q}$-norm defined by $\|\mathbf{z}\|_{\mathbf{Q}} = \mathbf{z}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{z}$ for a symmetric positive definite matrix $\mathbf{Q}$ that depends on β and k as well as on the covariance structure of $\mathbf{X}$. We describe how $\mathbf{Q}$ depends on β, k and $\mathbf{X}$ in Section 3.2. The minimization in (9) can be expressed as a posterior maximization problem given a conditional Gaussian likelihood over $\mathbf{y}$ and a Gaussian prior over $\boldsymbol{\phi}$. The posterior takes the form

$$p\,(\,\boldsymbol{\phi} \mid \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}\,) \propto \mathcal{N}(\mathbf{y}\,;\, \mathbf{X}\boldsymbol{\phi}, \mathbb{I})\, \mathcal{N}(\boldsymbol{\phi}\,;\, \boldsymbol{\theta}, \mathbf{Q}) \;\; . \tag{6}$$

Since $\boldsymbol{\phi}_{(k)}$ in (4) maximizes (6), we may conclude that k iterations of gradient descent in a linear regression model with squared error exactly computes the MAP estimate of $\boldsymbol{\phi}$, given a Gaussian-noised observation model and a Gaussian prior over $\boldsymbol{\phi}$ with parameters $\boldsymbol{\mu}_0 = \boldsymbol{\theta}$ and $\boldsymbol{\Sigma}_0 = \mathbf{Q}$. Therefore, in the linear case, MAML is exactly empirical Bayes using the MAP estimate as the point estimate of $\boldsymbol{\phi}$.

In the nonlinear case, MAML is again equivalent to an empirical Bayes procedure to maximize the marginal likelihood that uses a point estimate for $\boldsymbol{\phi}$ computed by one or a few steps of gradient descent. However, this point estimate is not necessarily the global mode of a posterior. We can instead understand the point estimate given by truncated gradient descent as the value of the local mode of an implicit posterior over $\boldsymbol{\phi}$ resulting from an empirical loss interpreted as a negative log likelihood, and regularization penalties and the early stopping procedure jointly acting as priors (Sjöberg & Ljung, 1995; Bishop, 1995; Duvenaud et al., 2016).

MAML can thus be understood as forming an approximate marginal negative log likelihood (NLL)

$$f_{\boldsymbol{\theta}}(\mathcal{T}_j) = -\log p\left(\, \mathbf{x}_{j_{N+1}}, \ldots \mathbf{x}_{j_{N+M}} \mid \hat{\boldsymbol{\phi}}_j \,\right)$$

for each task $\mathcal{T}_j$ using the point estimate

$$\hat{\boldsymbol{\phi}}_j = \boldsymbol{\theta} + \beta \nabla_{\boldsymbol{\theta}} \log p\left(\, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\theta}\,\right)$$

in the one-step case. The algorithm for MAML as probabilistic inference is given in Algorithm 2; Subroutine 3 computes each marginal NLL using a point estimate. Formulating MAML in this way, as probabilistic inference in a hierarchical Bayesian model, motivates the interpretation of using various optimization algorithms as a prior over task-specific parameters that we give in Section 3.2.

3.2 The Prior Over Task-Specific Parameters

By considering the case of a quadratic error function, we can better understand the role that early stopping has in defining a task-specific parameter prior $p(\,\boldsymbol{\phi} \mid \theta\,)$. Write the fast adaptation objective

as $\ell(\boldsymbol{\phi}) = -\log p(\, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\phi}\,)$, and consider a second-order approximation of the it objective about a minimum $\boldsymbol{\phi}^*$:

$$\ell(\boldsymbol{\phi}) \approx \tilde{\ell}(\boldsymbol{\phi}) := \|\boldsymbol{\phi} - \boldsymbol{\phi}^*\|^2_{\mathbf{H}^{-1}} + \ell(\boldsymbol{\phi}^*) \tag{7}$$

where the Hessian $\mathbf{H} = -\nabla^2_{\boldsymbol{\phi}}\, \ell(\boldsymbol{\phi})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}^*}$ is assumed to be positive definite so that $\tilde{\ell}$ is bounded below.

Furthermore, consider using a curvature matrix $\mathcal{B}$ to precondition the gradient in gradient descent, giving the update

$$\boldsymbol{\phi}_{(k)} = \boldsymbol{\phi}_{(k-1)} - \mathcal{B}\, \nabla_{\boldsymbol{\phi}_{(k-1)}}\, \ell(\boldsymbol{\phi}_{(k-1)}) \ . \tag{8}$$

If $\mathcal{B}$ is diagonal, we can identify (8) as a Newton method with a diagonal approximation to the inverse Hessian; using the inverse Hessian evaluated at the point $\boldsymbol{\phi}_{(k-1)}$ recovers Newton's method itself. On the other hand, learning the matrix $\mathcal{B}$ matrix via gradient descent provides a method to incorporate task-general information into the fast adaptation prior covariance. In particular, taking k steps of gradient descent from $\boldsymbol{\phi}_{(0)} = \boldsymbol{\theta}$ using the update rule in (8) gives a $\boldsymbol{\phi}_{(k)}$ that solves

$$\min \left(\|\boldsymbol{\phi} - \boldsymbol{\phi}^*\|^2_{\mathbf{H}^{-1}} + \|\boldsymbol{\phi}_{(0)} - \boldsymbol{\phi}\|^2_{\mathbf{Q}} \right) \ . \tag{9}$$

The minimization in (9) corresponds to taking a Gaussian prior $p(\, \boldsymbol{\phi} \mid \boldsymbol{\theta}\,)$ with mean $\boldsymbol{\theta}$ and covariance $\mathbf{Q}$ for $\mathbf{Q} = \mathbf{O}\boldsymbol{\Lambda}^{-1}((\mathbb{I} - \mathbf{B}\boldsymbol{\Lambda})^{-k} - \mathbb{I})\mathbf{O}^{\mathrm{T}}$ (Santos, 1996) where $\mathbf{B}$ is a diagonal matrix that results from a simultaneous diagonalization of $\mathbf{H}$ and $\mathcal{B}$ as $\mathbf{O}^{\mathrm{T}}\mathbf{H}\mathbf{O} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) = \boldsymbol{\Lambda}$ and $\mathbf{O}^{\mathrm{T}}\mathcal{B}^{-1}\mathbf{O} = \operatorname{diag}(b_1, \ldots, b_n) = \mathbf{B}$ with $b_i, \lambda_i \geq 0$ for $i = 1, \ldots, n$ (Theorem 8.7.1 in Golub & Van Loan, 1983). If the true objective is indeed quadratic, then, assuming the data is centred, $\mathbf{H}$ is the unscaled covariance matrix of features, $\mathbf{X}^{\mathrm{T}}\mathbf{X}$.

4 Improving Model-Agnostic Meta-Learning

Identifying MAML as a method for probabilistic inference in a hierarchical model allows us to develop novel improvements to the algorithm. In Section 4.1, we consider an approach from Bayesian parameter estimation to improve the MAML algorithm, and in Section 4.2, we discuss how to make this procedure computationally tractable.

4.1 Laplace's Method for Posterior Inference

We have shown that the MAML algorithm is an empirical Bayes procedure that employs a point estimate for the mid-level, task-specific parameters in a hierarchical Bayesian model. However, this assumption may lead to an innaccurate estimate of the integral in (2) if the joint density over the task-specific parameters and data, $p(\, \boldsymbol{\phi}_j, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_N} \mid \boldsymbol{\theta}\,)$, is not sharply peaked at the value of the point estimate. The Laplace approximation (Laplace, 1986; MacKay, 1992a;b) works to relax this assumption by replacing a point estimate of an integral with a Gaussian centered at a mode of the integrand, thereby forming a local quadratic approximation.

We can make use of this approximation to incorporate uncertainty about the task-specific parameters into the MAML algorithm at fast adaptation time. In particular, suppose that each integrand in (2) has a mode $\boldsymbol{\phi}_j^*$ at which it is locally well-approximated by a quadratic function. The Laplace approximation uses a second-order Taylor expansion of the negative log probability in order to approximate each integral in the sum in (2) as

$$\int p\left(\, \mathbf{X}_j \mid \boldsymbol{\phi}_j \,\right) p\left(\, \boldsymbol{\phi}_j \mid \boldsymbol{\theta} \,\right) \mathrm{d}\boldsymbol{\phi}_j \approx p\left(\, \mathbf{X}_j \mid \boldsymbol{\phi}_j^* \,\right) \det(\mathbf{H}/2\pi)^{-\frac{1}{2}}$$

where $\mathbf{H} = \nabla^2_{\boldsymbol{\theta}} \log p(\, \mathbf{X} \mid \boldsymbol{\phi}_j^*\,)$. Classically, the Laplace approximation uses the MAP estimate for $\boldsymbol{\phi}_j^*$, although any mode can be used as an expansion site. We use the point estimate $\hat{\boldsymbol{\phi}}$ uncovered by fast adaptation, in which case the MAML objective in (1) becomes

$$-\log p\left(\, \mathbf{X} \mid \boldsymbol{\theta} \,\right) \approx -\sum_j \left[\log p\left(\, \mathbf{X}_j \mid \boldsymbol{\phi}_j^* \,\right) + \log p\left(\, \boldsymbol{\phi}_j^* \mid \boldsymbol{\theta} \,\right) - \tfrac{1}{2} \log \det(\mathbf{H}) \right] \ .$$

We make use of the the Laplace approximation to replace Subroutine 3, which computes the task-specific marginal NLLs using a point estimate. We call this method Laplacian model-agnostic meta-learning (L-MAML), and give a replacement subroutine in Subroutine 4.

Subroutine ML−LAPLACE ($\boldsymbol{\theta}, \mathcal{T}$)

Draw N samples $\mathbf{x}_1, \ldots, \mathbf{x}_N \sim p_{\mathcal{T}}(\mathbf{x})$
Initialize $\boldsymbol{\phi} \leftarrow \boldsymbol{\theta}$
for k *in* $1, \ldots, K$ **do**
  Update $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \beta \nabla_{\boldsymbol{\phi}} \log p(\, \mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\phi}\,)$
end
Draw M samples $\mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \sim p_{\mathcal{T}}(\mathbf{x})$
Estimate quadratic curvature $\hat{\mathbf{H}}$ of $\log p(\, \mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \mid \boldsymbol{\phi}\,)$
return $-\log p(\, \mathbf{x}_{N+1}, \ldots, \mathbf{x}_{N+M} \mid \boldsymbol{\phi}\,) + \frac{1}{2} \log \det(\hat{\mathbf{H}})$

Subroutine 4: Subroutine for computing a Laplace approximation of the marginal likelihood.

4.2 Using Curvature Information to Improve MAML

A straightforward application of the Laplace approximation to a neural network model runs into several difficulties. The Hessian is intractable to compute exactly for all but the smallest models, and furthermore, is not guaranteed to be positive definite at all points, rendering the Laplace approximation undefined. To combat this, we instead seek an curvature matrix $\hat{\mathbf{H}}$ that approximates the quadratic curvature of a neural network objective function. Since it is well-known that the curvature associated with neural network objective functions is highly non-diagonal (Martens, 2016), a further requirement is that the matrix have off diagonal terms.

Martens & Grosse (2015) suggest using the Fisher information matrix (Fisher, 1925) as an approximation of curvature in natural gradient descent Amari (1998). A neural network with an appropriate choice of loss function is a probabilistic model and therefore defines a Fisher information matrix. Furthermore, the Fisher information matrix can be seen as a convex quadratic approximation to the objective function of a probabilistic neural model (Martens & Grosse, 2015). Importantly for our use case, the Fisher information matrix is positive definite by definition as well as non-diagonal and so is a suitable choice of curvature matrix for the Laplace approximation.

Martens & Grosse (2015) furthermore developed Kronecker-factored approximate curvature (K-FAC), a scheme for approximating the curvature of the objective function of a neural network with a block-diagonal approximation to the Fisher information matrix. Each block corresponds to a unique layer in the network, each block is further approximated as a Kronecker product (see Van Loan, 2000) of two much smaller matrices by assuming that the second-order statistics of the input activation and the back-propagated derivatives within a layer are independent. These two approximations ensure that the inverse of the Fisher information matrix can be computed efficiently for the purpose of natural gradient descent.

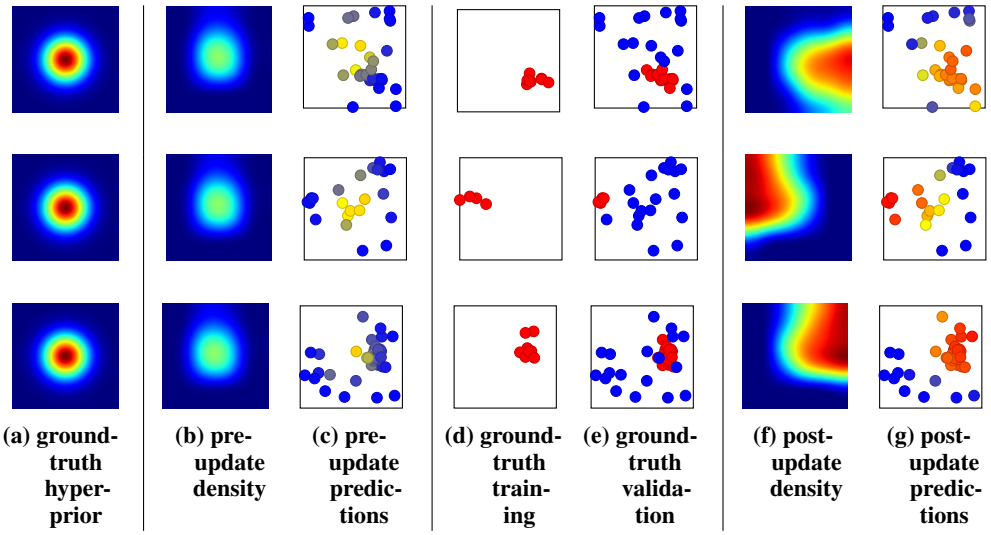
For the the Laplace approximation, we are interested in the determinant of a curvature matrix instead of its inverse. However, we may also make use of the approximations to the Fisher information matrix from K-FAC as well as properties of the Kronecker product. In particular, we use the fact that the determinant of a Kronecker product is the product of the exponentiated determinants of each of the factors, and that the determinant of a block diagonal matrix is the product of the determinants of the blocks. The determinants for each factor can be computed as efficiently as the inverses required by K-FAC, in $\mathcal{O}(n^3)$ time.

5 Experimental Evaluation

5.1 Warmup: Toy Nonlinear Model

The equivalence between MAML and hierarchical Bayes suggests that we should expect MAML to behave like an algorithm that learns the mean of a Gaussian prior on model parameters, and uses this prior as an initialization during fast adaptation. To illustrate this relationship between MAML and hierarchical Bayes, we present a toy example that demonstrates how MAML can recover the form of a hierarchical probabilistic model that generates a toy dataset in $d = 2$ dimensions.

Consider a graphical model consisting of a Normal-Inverse-Wishart hyperprior, with hyperparameters $\lambda \in \mathbb{R}, \nu \in \mathbb{R}, \boldsymbol{\mu}_0 \in \mathbb{R}^D, \boldsymbol{\Psi} \in \mathbb{R}^{d \times d}$, over the mean $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}$ of a multivariate Normal observation model, where the data is distributed according to $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,



(a) ground-truth hyper-prior | **(b) pre-update density** | **(c) pre-update predic-tions** | **(d) ground-truth train-ing** | **(e) ground-truth valida-tion** | **(f) post-update density** | **(g) post-update predic-tions**

Figure 5: A visualization of the sigmoid output layer of a feedforward neural network with one hidden layer, trained with MAML for 2000 iterations to estimate a decision boundary (Figure 5f) for a novel class from only positive (red) examples of the class (Figure 5d). Each row corresponds to a new class (or task) sampled from the hyperprior. Note that MAML has uncovered expectations of the marginal prior mean and covariance (compare Figure 5a, Figure 5b), and taking a gradient step moves the density to the expected mean and covariance of the task density (Figure 5f).

and the mean and covariance are distributed according to $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \text{NIW}(\boldsymbol{\mu}_0, \lambda, \boldsymbol{\Psi}, \nu)$. We may generate observations from this model by sampling $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from the inverse-Wishart distribution, and then sampling an observation $\mathbf{x}$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In the context of the MAML algorithm, sampling $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ corresponds to sampling a task from the task distribution $p_{\mathscr{D}}(\mathcal{T})$, which is parameterized by $\boldsymbol{\mu}_0$, λ, $\boldsymbol{\Psi}$, and ν.

From the experiment in Figure 5, we can see that MAML recovers an estimate of the parameters that define the underlying generative model of the data. In particular, MAML uncovers expectations of the marginal prior mean and covariance, and taking a gradient step moves the density to the expected mean and covariance of the task density. This qualitative result affirms the connection bewteen MAML and hierarchical Bayesian inference.

5.2 Large-Scale Experiment: *mini*ImageNet

We evaluate on the miniImageNet Ravi & Larochelle (2017) 1-shot, 5-way classification task, a standard benchmark in few-shot classification. *mini*ImageNet comprises 64 training classes, 12 validation classes, and 24 test classes. Following the setup of Vinyals et al. (2016), we structure the N-shot, J-way classification task as follows: The model observes N instances of J unseen classes, and is evaluated on its ability to classify M new instances within the J classes.

We use a neural network architecture standard to few-shot classification (*e.g.,* Vinyals et al., 2016; Ravi & Larochelle, 2017), consisting of 4 layers with 3×3 convolutions and 64 filters, followed by batch normalization (BN) (Ioffe & Szegedy, 2015), a ReLU nonlinearity, and 2×2 max-pooling. For the scaling variable β and centering variable γ of BN (see Ioffe & Szegedy, 2015), we ignore the fast adaptation update as well as the Fisher factors for the the computation of the approximate Fisher information matrix . We use Adam (Kingma & Ba, 2014) as the meta-optimizer, and standard gradient descent with a fixed learning rate to update the model during fast adaptation. L-MAML requires an additional parameter that weights the regularization term $\log\det\hat{\mathbf{H}}$ contributed by the the Laplace approximation. We chose a weight of 10^{-6} via cross-validation.

We find that L-MAML is practical enough to be applied to this larger-scale problem. In particular, our TensorFlow implementation of MAML trains for 60,000 iterations on one TITAN Xp GPU in under 5 hours, compared to 9 hours to train L-MAML due to the additional computational overhead from K-FAC. As shown in Table 1, L-MAML achieves comparable performance to the state-of-the-art meta-learning method by Triantafillou et al. (2017). While the gap between MAML and L-MAML is

Model	1-shot acc.
Fine-tuning[*]	28.86 ± 0.54
Nearest Neighbor[*]	41.08 ± 0.70
Matching Networks FCE (Vinyals et al., 2016)[*]	43.56 ± 0.84
Meta-Learner LSTM (Ravi & Larochelle, 2017)[*]	43.44 ± 0.77
Prototypical Networks (Snell et al., 2017)[**]	46.61 ± 0.78
mAP-DLM (Triantafillou et al., 2017)	$\mathbf{49.82 \pm 0.78}$
MAML (Finn et al., 2017)	48.70 ± 1.84
L-MAML (Ours)	$\mathbf{49.40 \pm 1.83}$

Table 1: One-shot classification performance on miniImageNet for methods that report results using the standard four-layer convolutional embedding model of Vinyals et al. (2016).[1] All results are averaged over 600 test episodes, and we report 95% confidence intervals. [*]Results reported by (Ravi & Larochelle, 2017) on a standardized dataset. [**]We report test accuracy for models matching train and test "shot" and "way".

small, the improvement from the Laplace approximation suggests that more accurate approximations to the marginalization over task-specific parameters may lead to further improvements.

6 Related Work

Approaches to meta-learning have come from two largely disjoint communities. Machine learning and deep learning researchers have addressed few-shot learning either through the design of network architectures that ingest the few-shot training samples directly (*e.g.,* Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Hariharan & Girshick, 2017; Triantafillou et al., 2017), or formulating the problem as one of *learning to learn*, or *meta-learning* (*e.g.,* Schmidhuber, 1987; Bengio et al., 1991; Schmidhuber, 1992; Bengio et al., 1992). Meta-learning has recently been implemented with learned update rules (*e.g.,* Andrychowicz et al., 2016; Ravi & Larochelle, 2017; Li & Malik, 2017a; Ha et al., 2017)) and memory-augmentation (*e.g.,* Santoro et al., 2016)). In this work, we build on the gradient-based meta-learning method proposed by Finn et al. (2017).

Few-shot learning also has a long history in hierarchical Bayesian modeling (*e.g.,* Tenenbaum, 1999; Fei-Fei et al., 2003; Lawrence & Platt, 2004; Yu et al., 2005; Gao et al., 2008; Daumé III, 2009; Wan et al., 2012). A related subfield is that of transfer learning, which has used hierarchical Bayes extensively (*e.g.,* Raina et al., 2006). A variety of inference methods have been used in Bayesian models, including exact inference (Lake et al., 2011), sampling methods (Salakhutdinov et al., 2012), variational methods (Edwards & Storkey, 2017).

Our work bridges the gap between gradient-based meta-learning methods and hierarchical Bayesian modeling. Our contribution is not to formulate the meta-learning problem as a hierarchical Bayesian model, but instead to formulate a gradient-based meta-learner as hierarchical Bayesian inference, thus providing a way to efficiently perform posterior estimation in a model agnostic manner.

7 Conclusion

We have shown that model-agnostic meta-learning (MAML) estimates the parameters of a prior distribution in a hierarchical Bayesian model. By casting gradient-based meta-learning within a Bayesian framework, our analysis opens the door to a number of novel improvements inspired by probabilistic machinery.

As a step in this direction, we propose an extension to MAML that employs a Laplace approximation to the posterior distribution over task-specific parameters. This technique provides a more accurate estimate of the integral that was originally approximated as a point estimate during fast adaptation. We show how to approximate the quantity required by the Laplace approximation using Kronecker-factored approximate curvature (K-FAC), a method recently proposed to approximate the quadratic curvature of a neural network objective function for the purpose of second-order gradient descent.

[1] We omit the performance of TCML (Mishra et al., 2017) from the table as their ResNet embedding architecture has significantly more parameters than the other methods and is thus not comparable. Their 1-shot performance is 55.71 ± 0.99.

Our contribution illuminates the road to exploring further connections between gradient-based meta-learning methods and hierarchical Bayesian modelling. For instance, in this work we assume that the predictive distribution over new datapoints is narrow and well-approximated by a point estimate. We may instead employ methods that make use of the variance of the distribution over task-specific parameters in order to model the predictive density over examples from a novel task.

Furthermore, it is known that the Laplace approximation is inaccurate in cases where the integral is highly skewed, or is not unimodal and thus is not amenable to approximation by a single Gaussian mode. This could be solved by using a finite mixture of Gaussians, which can approximate many density functions arbitrarily well (Sorenson & Alspach, 1971; Alspach & Sorenson, 1972).

REFERENCES

Daniel Alspach and Harold Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 1972.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29*, pp. 3981–3989, 2016.

Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In *Proceedings of the Conference on Optimality in Artificial and Biological Neural Networks*, pp. 6–8, 1992.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. Learning a synaptic learning rule. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2, 1991.

J.M. Bernardo and A.F.M. Smith. *Bayesian Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons Canada, Limited, 2006. ISBN 9780470028735.

Christopher M Bishop. Regularization and complexity control in feed-forward networks. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 141–148, 1995.

Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

Hal Daumé III. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 135–142. AUAI Press, 2009.

David Duvenaud, Dougal Maclaurin, and Ryan Adams. Early stopping as nonparametric variational inference. In *Artificial Intelligence and Statistics*, pp. 1070–1077, 2016.

Harrison Edwards and Amos Storkey. Towards a neural statistician. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Li Fei-Fei et al. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th Conference on Computer Vision and Pattern Recognition*, pp. 1134–1141. IEEE, 2003.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Ronald Aylmer Fisher. Theory of statistical estimation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 22, pp. 700–725. Cambridge University Press, 1925.

Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 283–291. ACM, 2008.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014.

Gene H. Golub and Charles F. Van Loan. *Matrix computations*. The Johns Hopkins University Press, 1983.

David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Bharath Hariharan and Ross Girshick. Low-shot visual object recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Gregory Koch. Siamese neural networks for one-shot image recognition. Master's thesis, University of Toronto, 2015.

Brenden M Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society*, 2011.

Pierre Simon Laplace. Memoir on the probability of the causes of events. *Statistical Science*, 1(3):364–378, 1986.

Neil D Lawrence and John C Platt. Learning to learn with the informative vector machine. In *Proceedings of the 21 st International Conference on Machine Learning (ICML)*, pp. 65, 2004.

Ke Li and Jitendra Malik. Learning to optimize. In *Proceedings of the 5th International Conference on Learning Representations*, 2017a.

Ke Li and Jitendra Malik. Learning to optimize neural nets. In *Proceedings of the 34th International Conference on Machine Learning*, 2017b.

David JC MacKay. The evidence framework applied to classification networks. *Neural computation*, 4(5): 720–736, 1992a.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3): 448–472, 1992b.

James Martens. *Second-order optimization for neural networks*. PhD thesis, 2016.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pp. 2408–2417, 2015.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. Meta-learning with temporal convolutions. *arXiv preprint arXiv:1707.03141*, 2017.

Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 1, pp. 437–442. IEEE, 1992.

Rajat Raina, Andrew Y Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 713–720. ACM, 2006.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pp. 195–206. PMLR, 2012.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*, pp. 1842–1850, 2016.

Reginaldo J. Santos. Equivalence of regularization and truncated iteration for general ill-posed problems. *Linear Algebra and its Applications*, 236(15):25–33, 1996. doi: http://dx.doi.org/10.1016/0024-3795(94)00114-6.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning*. PhD thesis, Institut für Informatik, Technische Universität München, 1987.

Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

Jonas Sjöberg and Lennart Ljung. Overtraining, regularization and searching for a minimum, with application to neural networks. *International Journal of Control*, 62(6):1391–1407, 1995.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30*, 2017.

Harold W Sorenson and Daniel L Alspach. Recursive bayesian estimation using gaussian sums. *Automatica*, 7 (4):465–479, 1971.

Joshua B Tenenbaum. *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology, 1999.

Sebastian Thrun and Lorien Pratt. Learning to learn. 1998.

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. *arXiv preprint arXiv:1707.02610*, 2017.

Charles F Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1):85–100, 2000.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29*, pp. 3630–3638, 2016.

Jing Wan, Zhilin Zhang, Jingwen Yan, Taiyong Li, Bhaskar D Rao, Shiaofen Fang, Sungeun Kim, Shannon L Risacher, Andrew J Saykin, and Li Shen. Sparse bayesian multi-task learning for predicting cognitive outcomes from neuroimaging measures in alzheimer's disease. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 940–947. IEEE, 2012.

Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. *arXiv preprint arXiv:1703.04813*, 2017.

Kai Yu, Volker Tresp, and Anton Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine learning (ICML)*, pp. 1012–1019, 2005.