Empirical Bayes Meta-Learning with Synthetic Gradients

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We revisit the hierarchical Bayes and empirical Bayes formulations for multi-task learning, which can naturally be applied to meta-learning. The evidence lower bound of the marginal log-likelihood of empirical Bayes decomposes as a sum of local KL divergences between the variational posterior and the true posterior of each task. We derive an amortized variational inference that couples all the variational posteriors into a meta-model, which consists of a synthetic gradient network and an initialization network. Our empirical results on the mini-ImageNet benchmark for episodic few-shot classification significantly outperform previous state-of-the-art methods.

1 A Bayesian formulation for meta-learning

We consider a multi-task setting in which we have a collection of datasets $\mathcal{D} := \{d_i\}_{i=1}^N$, where d_i is the data associated with task i. Unlike traditional machine learning problems, $\mathcal{D}$ is a dataset of datasets. We further assume that $d_1, \dots d_N$ are iid samples drawn from the empirical distribution $\hat{p}_{\mathcal{D}}(d) = \frac{1}{N}\sum_{i=1}^N \delta(d - d_i)$. The goal of this problem is to leverage the characteristics shared by all tasks, such that the learning of unseen tasks is sample-efficient. Later in the paper we will split $\mathcal{D}$ into three subsets ($\mathcal{D}^{\text{train}}$, $\mathcal{D}^{\text{val}}$ and $\mathcal{D}^{\text{test}}$). We will train a meta-model on $\mathcal{D}^{\text{train}}$, select the best hyper-parameters on $\mathcal{D}^{\text{val}}$, and finally report the performance on $\mathcal{D}^{\text{test}}$.

If we consider for now that each dataset is composed of iid input-output pairs: $d_i := \{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$, then the log-likelihood takes the form

$$\begin{aligned} \log p(d_i|w_i) &= \sum_{j=1}^{n_i} \log p(y_{ij}|x_{ij}, w_i) + \log p(x_{ij}|w_i), \\ &= \sum_{j=1}^{n_i} -\ell_i\big(y_{ij}, \hat{y}_{ij}(x_{ij}, w_i)\big) + \text{ constant}, \end{aligned} \tag{1}$$

where w_i and ℓ_i are task-specific parameter and loss respectively for task i; $\hat{y}_{ij}$ is a prediction of y_{ij}. For example, ℓ_i may involve a relabeling of y_{ij} as the case of few-shot classification [Vinyals et al., 2016]. Note that we treat $\log p(x_{ij}|w_i)$ as a constant since it is irrelevant for predicting y_{ij}. Notation-wise, we will also use $x_i = \{x_{ij}\}_{j=1}^{n_i}$ and $y_i = \{y_{ij}\}_{j=1}^{n_i}$.

Since we are modeling a distribution over tasks, a natural way to formulate this problem is to introduce a distribution over the task-specific parameter, namely, $p(w_i|\psi)$ with the hyper-parameter ψ shared across all tasks, and to either consider a *hierarchical Bayes* formulation

$$p(\mathcal{D}) = \int_\psi p(\mathcal{D}|\psi)p(\psi) = \int_\psi \Big[\prod_{i=1}^N \int_{w_i} p(d_i|w_i)p(w_i|\psi)\Big]p(\psi) \tag{2}$$

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

or an *empirical Bayes* formulation based on the *type-II likelihood*

$$p_\psi(\mathcal{D}) = \prod_{i=1}^{N} p_\psi(d_i) = \prod_{i=1}^{N} \int_{w_i} p(d_i|w_i) p_\psi(w_i), \tag{3}$$

in which ψ is estimated in a frequentist sense by solving $\max_\psi \log p_\psi(\mathcal{D})$. In other words, we approximate the posterior on the hyper-parameter ψ with a point estimate.

2 Variational inference with synthetic gradients

In this work, we consider variational inference for the empirical Bayes formulation (3). Specifically, by introducing a variational distribution $q_{\theta_i}(w_i)$ for each task i with parameter θ_i, we have the following *evidence lower bound* (ELBO) on the marginal log-likelihood

$$\begin{aligned}\log p_\psi(\mathcal{D}) &\geq \sum_{i=1}^{N} \int_{w_i} q_{\theta_i}(w_i) \log \frac{p(d_i|w_i) p_\psi(w_i)}{q_{\theta_i}(w_i)} \\ &= \sum_{i=1}^{N} \Big[\mathbb{E}_{w_i \sim q_{\theta_i}} \big[\log p(d_i|w_i) \big] - D_{\mathrm{KL}}\big(q_{\theta_i}(w_i) \| p_\psi(w_i) \big) \Big]. \end{aligned} \tag{4}$$

Maximizing the ELBO in (4) with respect to $\theta_1, \ldots, \theta_N$ and ψ is equivalent to

$$\min_\psi \min_{\theta_1,\ldots,\theta_N} \sum_{i=1}^{N} D_{\mathrm{KL}}\Big(q_{\theta_i}(w_i) \| p_\psi(w_i|d_i) \Big), \tag{5}$$

where $p_\psi(w_i|d_i) = \frac{p(d_i|w_i)p_\psi(w_i)}{\int_{w_i} p(d_i|w_i)p_\psi(w_i)}$ is the true posterior induced by the prior $p_\psi(w_i)$ and the likelihood $p(d_i|w_i)$.

The optimization in (5) becomes more and more costly as N increases. We wish to bypass this computationally expensive optimization and to take advantage of the fact that individual KL divergences indeed share the same structure. To this end, instead of introducing N different variational distributions, we consider a commonly parameterized family of distributions, which is defined implicitly by a deep neural network ϕ taking as input d_i. Replacing each q_{θ_i} by $q_{\phi(d_i)}$, (5) can be written as

$$\min_\psi \min_\phi \sum_{i=1}^{N} D_{\mathrm{KL}}\Big(q_{\phi(d_i)}(w_i) \| p_\psi(w_i|d_i) \Big), \tag{6}$$

which is also known as *amortized variational inference* in the literature [Kingma and Welling, 2013, Rezende et al., 2014].

It is however not trivial to design a network architecture to implement $\phi(d_i)$ since d_i is itself a dataset. A common strategy [Garnelo et al., 2018] is to aggregate the information from all individual examples via a permutation invariant function. However, as pointed out by Kim et al. [2019], such a strategy tends to underfit d_i, because the aggregation does not necessarily attain the most relevant information for producing w_i. We instead focus on the optimization aspect of q_{θ_i}. Consider a gradient descent on θ_i for optimizing (5)

$$\theta_i^{t+1} = \theta_i^t - \eta \, \nabla_{\theta_i} D_{\mathrm{KL}}\Big(q_{\theta_i}(w_i) \| p_\psi(w_i|d_i) \Big). \tag{7}$$

We would like to parameterize this optimization dynamics up to the T-th step via $\phi(d_i)$. It consists of parameterizing

(a) the gradient $\nabla_{\theta_i} D_{\mathrm{KL}}(q_{\theta_i}(w_i) \| p_\psi(w_i|d_i))$;

(b) the initialization θ_i^0.

By doing so, θ_i^T becomes a function of ϕ and d_i, we therefore realize $q_{\phi(d_i)}$ as $q_{\theta_i^T}$.

For (a), we observe that

$$\nabla_{\theta_i} D_{\mathrm{KL}}\Big(q_{\theta_i}(w_i) \| p_\psi(w_i|d_i) \Big) = \mathbb{E}_\epsilon \Big[\sum_{j=1}^{n_i} \frac{\partial \ell_i}{\partial \hat{y}_{ij}} \frac{\partial \hat{y}_{ij}}{\partial w_i} \frac{\partial w_i}{\partial \theta_i} \Big] + \nabla_{\theta_i} D_{\mathrm{KL}}\Big(q_{\theta_i}(w_i) \| p_\psi(w_i) \Big) \tag{8}$$

under a reparameterization $w_i = w_i(\theta_i, \epsilon)$ with $\epsilon \sim p(\epsilon)$. All the terms in (8) can be computed without the groundtruth label y_{ij} except for $\frac{\partial \ell_i}{\partial \hat{y}_{ij}}$, thus, we introduce a deep neural network $\xi(\hat{y}_{ij})$ to synthesize it. The idea of synthetic gradients [Jaderberg et al., 2017] was originally proposed to parallelize the back-propagation. Here, the purpose of $\xi(\hat{y}_{ij})$ is to update θ_i regardless of the groundtruth labels, which is slightly different from its original purpose. Besides, we do not introduce an additional loss to force $\xi(\hat{y}_{ij})$ to approximate $\frac{\partial \ell_i}{\partial \hat{y}_{ij}}$ since $\xi(\hat{y}_{ij})$ will be learned to yield a reasonable θ_i^T even without mimicking the true gradient.

For (b), we can either let $\theta_i^0 = \lambda$ to be data-independent with a global learnable initialization λ or let $\theta_i^0 = \lambda(x_i)$ such that $\lambda(\cdot)$ is a permutation invariant mapping from x_i to θ_i^0. If in addition we are given a *support set* $d_i^{\text{supp}} := \{x_{ij}^{\text{supp}}, y_{ij}^{\text{supp}}\}_{j=1}^{n_i'}$ for task i[1], a better initialization can be computed via $\theta_i^0 = \lambda(d_i^{\text{supp}})$.

To sum up, we have derived a particular implementation of $\phi(d_i)$ inspired by (7), such that $\phi \equiv (\lambda, \xi)$. Specifically, we have $\phi(d_i) = \phi(x_i) = \theta_i^T$, which is computed via

$$\theta_i^{t+1} = \theta_i^t - \eta \left[\mathbb{E}_\epsilon \left[\textstyle\sum_{j=1}^{n_i} \xi(\hat{y}_{ij}) \frac{\partial \hat{y}_{ij}}{\partial w_i} \frac{\partial w_i}{\partial \theta_i} \right] + \nabla_{\theta_i} D_{\text{KL}}\Big(q_{\theta_i}(w_i) \| p_\psi(w_i) \Big) \right] \text{ with } \theta_i^0 = \lambda(x_i). \quad (9)$$

The fundamental reason we use a parameteric update to obtain θ_i^T rather than follow (7) is because we do not have access to y_i when testing on unseen tasks from the test-set $\mathcal{D}^{\text{test}}$. The same issue occurs in supervised learning with VAE. For instance, if we were following the conditional VAE [Sohn et al., 2015], for training, we would sample w_i from $q_{\phi(d_i)}$, and, for testing, we would sample w_i either from p_ψ or from an iteratively estimated variational posterior starting from a random prediction of y_i. Although this gives a valid solution, the way to sample w_i would be inconsistent for training and testing, which would render the variational inference suboptimal.

Model specification In (6), there are two parameteric models to be learned: $q_{\phi(d_i)}$ and p_ψ. To obtain a closed-form KL term, we restrict ourselves to Gaussian models[2], such that both $q_{\phi(d_i)}$ and p_ψ are Gaussian distributions with diagonal covariance. In addition, we may introduce a parameterized feature module f to enhance the likelihood model $p_f(y_{ij}|x_{ij}, w_i)$ such that

$$-\log p_f(y_{ij}|x_{ij}, w_i) = \ell_i\big(y_{ij}, \hat{y}_{ij}(f(x_{ij}), w_i)\big). \quad (10)$$

Following Gidaris and Komodakis [2018], Qiao et al. [2018], we implement $f(\cdot)$ by a 4-layer convolutional network or a wide ResNet (WRN-28-10) [Zagoruyko and Komodakis, 2016].

3 Few-shot classification on mini-ImageNet

We evaluate our method on the mini-ImageNet dataset, which is an episodic few-shot classification benchmark proposed by Vinyals et al. [2016]. An episode/task i consists of a *query set* d_i and a *support set* d_i^{supp}. When we say an episode i is k*-way-*n*-shot* we mean that d_i^{supp} is formed by first sampling k categories from a pool of categories; then, for each sampled category, n examples are drawn and a new label taken from $\{0, \ldots, k-1\}$ is assigned to these examples. The goal of this problem is to predict the labels of the query set, which are provided as ground truth during training.

The mini-ImageNet dataset contains 100 different categories with 600 images per category, each of size 84×84 pixels. We used the splits by Ravi and Larochelle [2016] that include 64 categories to form $\mathcal{D}^{\text{train}}$, 16 categories to form $\mathcal{D}^{\text{val}}$, and 20 categories to form $\mathcal{D}^{\text{test}}$.

Following Gidaris and Komodakis [2018], we pretrain the feature network $f(\cdot)$ on $\mathcal{D}^{\text{train}}$ for standard 64-way classification. We also reuse their feature averaging network as our initialization network $\lambda(\cdot)$, which basically averages the feature vectors of all data points from the same category and then scale each feature dimension differently by a learned coefficient. For the gradient network $\xi(\cdot)$, we implement a three-layer MLP with hidden-layer size $8k$. Finally, for the predictor $\hat{y}_{ij}(\cdot, w_i)$, we adopt

[1]This setting is called *few-shot learning* since n_i' is in general small. For a special case where $n_i' = 0$, the setting is called *zero-shot learning*.

[2]It is however possible to consider more powerful parameterization. For example, implementing the prior $p_\psi(w_i)$ by PixelCNN [Van den Oord et al., 2016] with lossy compression similar to that of VQ-VAE2 [Razavi et al., 2019]. We leave that for future work.

the cosine-similarity based classifier advocated by Chen et al. [2019] and Gidaris and Komodakis [2018].

There are two types of evaluation: (a) the standard k-way few-shot classification proposed by Vinyals et al. [2016] and (b) the learning without forgetting (LwoF) few-shot classification proposed by Gidaris and Komodakis [2018]. We use the same evaluation code provided by Gidaris and Komodakis [2018]. For (b), we additionally evaluate the performance on the 64 base categories as a $(64+5)$-way classification. In order to classify base categories, we implement p_ψ as a mixture of Gaussians with 64 components and equal mixing coefficients. The weight of the predictor for classifying base categories are sampled from p_ψ. Note that the KL terms can still be computed in closed form.

For training, we use ADAM with batch size 8 for 60 epochs, where the initial learning rate is 10^{-3} and dropped by a factor 0.1 at epoch 10, 25, 50. We use the validation set $\mathcal{D}^{\text{val}}$ to select the best performing model and then use it to test on the test-set $\mathcal{D}^{\text{test}}$.

In Table 1 and Tabel 2 we show a comparison between the state-of-the-art approaches and several variants of our method (varying T or $f(\cdot)$) on $\mathcal{D}^{\text{val}}$ and $\mathcal{D}^{\text{test}}$ respectively. We observe that our methods yield a clear performance boost on novel categories, especially when evaluated on the standard few-shot classification setting. Comparing the cases $T=0$ and $T=5$, there are clear $>4\%$ and $>10\%$ improvements with CNN feature networks, which becomes even more significant with WRN-28-10 features.

Methods	5-way-5-shot			5-way-1-shot		
	Novel	Base	Both	Novel	Base	Both
Vinyals et al. [2016]	$68.87 \pm 0.38\%$	-	-	$55.53 \pm 0.48\%$	-	-
Snell et al. [2017]	$72.67 \pm 0.37\%$	62.10%	32.70%	$54.44 \pm 0.48\%$	52.35%	26.68%
Gidaris and Komodakis [2018]	$74.92 \pm 0.36\%$	**70.88%**	60.50%	$58.55 \pm 0.50\%$	**70.73%**	50.50%
Standard few-shot classification						
Ours $T=0$	$73.18 \pm 0.34\%$	-	-	$55.42 \pm 0.44\%$	-	-
Ours $T=1$	$76.09 \pm 0.35\%$	-	-	$60.74 \pm 0.50\%$	-	-
Ours $T=3$	$77.53 \pm 0.35\%$	-	-	$65.14 \pm 0.54\%$	-	-
Ours $T=5$	$\mathbf{77.74 \pm 0.36\%}$	-	-	$\mathbf{66.04 \pm 0.59\%}$	-	-
LwoF few-shot classification						
Ours $T=0$	$73.13 \pm 0.34\%$	70.51%	58.09%	$55.22 \pm 0.45\%$	70.01%	47.56%
Ours $T=1$	$76.69 \pm 0.34\%$	70.40%	**62.10%**	$61.81 \pm 0.50\%$	70.09%	53.53%
Ours $T=3$	$76.54 \pm 0.35\%$	69.30%	60.91%	$63.92 \pm 0.54\%$	70.19%	**54.89%**
Ours $T=5$	$76.68 \pm 0.35\%$	70.28%	61.93%	$64.39 \pm 0.58\%$	69.88%	54.65%

Table 1: Average classification accuracies on the **validation set** of mini-ImageNet. The "Novel" columns report the average 5-way and 1-shot or 5-shot classification accuracies of novel classes (with 95% confidence intervals), the "Base" and "Both" columns report the classification accuracies of base classes and of both type of classes respectively. In order to report those results we sampled 2000 tasks each with $15 \times k$ test examples of novel classes and $15 \times k$ test examples of base classes.

Methods	5-way-5-shot			5-way-1-shot		
	Novel	Base	Both	Novel	Base	Both
Vinyals et al. [2016]	55.30%	-	-	43.60%	-	-
Ravi and Larochelle [2016]	$60.20 \pm 0.71\%$	-	-	$43.40 \pm 0.77\%$	-	-
Finn et al. [2017]	$63.10 \pm 0.92\%$	-	-	$48.70 \pm 1.84\%$	-	-
Snell et al. [2017]	$68.20 \pm 0.66\%$	-	-	$49.42 \pm 0.78\%$	-	-
Mishra et al. [2017]	$68.88 \pm 0.92\%$	-	-	$55.71 \pm 0.99\%$	-	-
Gidaris and Komodakis [2018]	$73.00 \pm 0.64\%$	**70.90%**	59.35%	$55.95 \pm 0.84\%$	**70.72%**	49.08%
Standard few-shot classification						
Ours $T=0$	$71.48 \pm 0.64\%$	-	-	$53.62 \pm 0.79\%$	-	-
Ours $T=1$	$74.12 \pm 0.63\%$	-	-	$58.74 \pm 0.89\%$	-	-
Ours $T=3$	$75.43 \pm 0.67\%$	-	-	$62.59 \pm 1.02\%$	-	-
Ours $T=5$	$\mathbf{75.73 \pm 0.71\%}$	-	-	$\mathbf{63.26 \pm 1.07\%}$	-	-
Ours $T=3$ and $f=$ WRN-28-10	$\mathbf{78.92 \pm 0.37\%}$	-	-	$\mathbf{67.92 \pm 0.55\%}$	-	-
LwoF few-shot classification						
Ours $T=0$	$70.93 \pm 0.63\%$	69.46%	56.79%	$54.43 \pm 0.76\%$	69.30%	47.85%
Ours $T=1$	$74.42 \pm 0.66\%$	69.28%	**60.20%**	$60.35 \pm 0.88\%$	69.10%	52.52%
Ours $T=3$	$73.86 \pm 0.66\%$	68.27%	58.71%	$62.02 \pm 0.93\%$	69.45%	**53.52%**
Ours $T=5$	$74.10 \pm 0.67\%$	69.06%	59.74%	$61.82 \pm 1.00\%$	68.80%	52.95%

Table 2: Average classification accuracies on the **test set** of mini-ImageNet. In order to report those results we sampled 600 tasks in a similar fashion as for the validation set of mini-ImageNet (see Table 1).

References

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

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-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1627–1635. JMLR, 2017.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7229–7238, 2018.

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

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 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 29*, pages 3630–3638. Curran Associates, Inc., 2016.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.