BNAIC/BeNeLearn 2022

November 8, 2022

Ozgur Taylan Turan

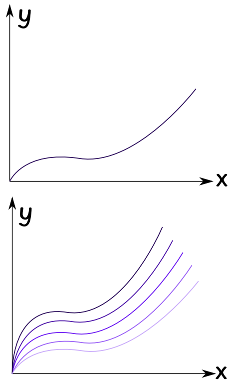
When MAML Learns Quickly, Does It Generalize Well?

O. Taylan Turan[1], David M.J. Tax[1], and Marco Loog[1]

[1] Delft University of Technology, Pattern Recognition and Bioinformatics Laboratory

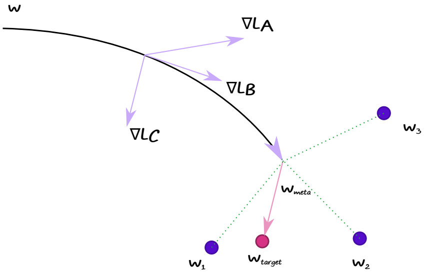
Meta-Learning

- Introduced in 90s
- Leverage different learning problems for a target problem
- Especially useful in few-shot learning



How Does MAML[1] Work?

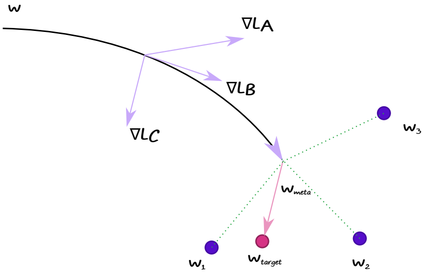
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps (quick adaptation)



[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

How Does MAML[1] Work?

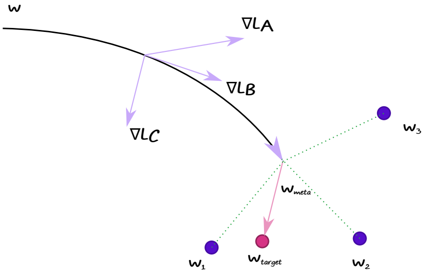
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps (quick adaptation)



[1] C. Finn, P. Abbeel, and S. Levine (July 2017). “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

How Does MAML[1] Work?

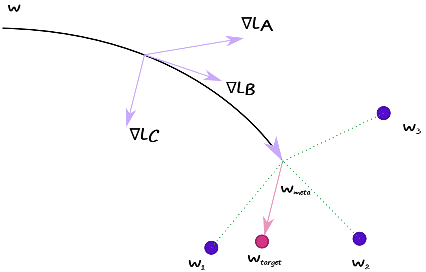
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps (quick adaptation)



[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

How Does MAML[1] Work?

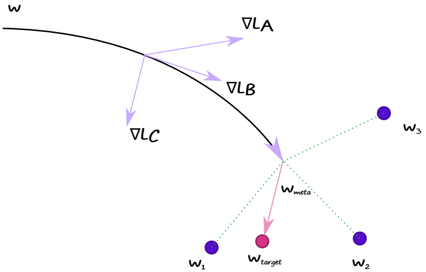
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps (quick adaptation)



[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

How Does MAML[1] Work?

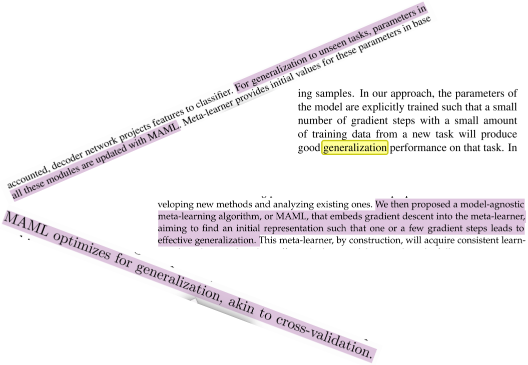
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps (quick adaptation)



[1] C. Finn, P. Abbeel, and S. Levine (July 2017). "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In: *arXiv:1703.03400 [cs]*. arXiv: 1703.03400 [cs]

What is the problem?

- Generalization
- Quick adaptation is not needed by many settings
- Robotics/image classification/regression applications



accounted, decoder network projects features to classifier. For generalization to unseen tasks, parameters in all these modules are updated with MAML. Meta-learner provides initial values for these parameters in base

ing samples. In our approach, the parameters of the model are explicitly trained such that a small number of gradient steps with a small amount of training data from a new task will produce good generalization performance on that task. In

veloping new methods and analyzing existing ones. We then proposed a model-agnostic meta-learning algorithm, or MAML, that embeds gradient descent into the meta-learner, aiming to find an initial representation such that one or a few gradient steps leads to effective generalization. This meta-learner, by construction, will acquire consistent learn-

MAML optimizes for generalization, akin to cross-validation.

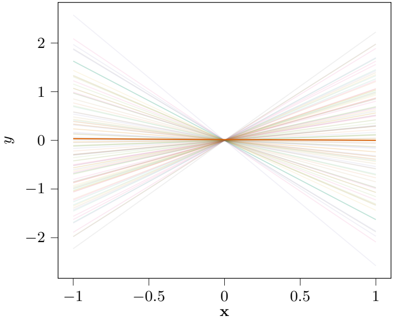
AIM

Investigate the effect of gradient step limitation on generalization performance!

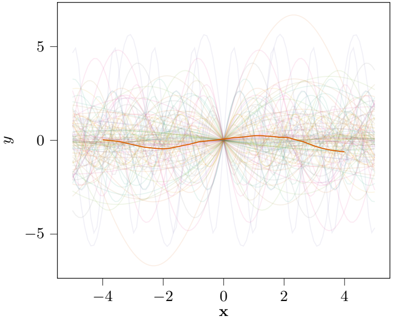
Experimental Setup

- Tasks: linear/nonlinear regression problems with noisy ($\varepsilon \sim \mathcal{N}(0, \sigma^2)$) observations of functions $f(\mathbf{x})$

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon \tag{1}$$

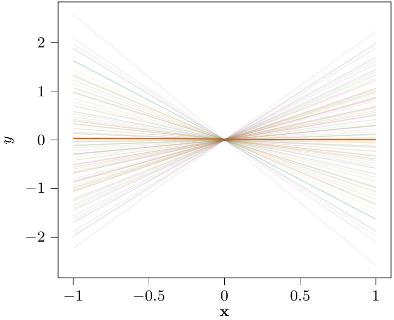


$f(\mathbf{x}) := \mathbf{x}^\mathsf{T}\mathbf{a}$

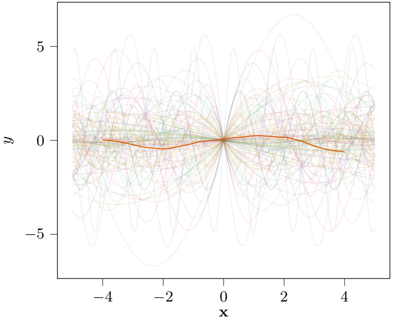


$f(\mathbf{x}) := \sin(\mathbf{x} + \boldsymbol{\phi})^\mathsf{T}\mathbf{a}$

Experimental Setup



$$f(\mathbf{x}) := \mathbf{x}^\top \mathbf{a}$$
$$\mathbf{a} \sim \mathcal{N}(m\mathbf{1}, c\mathbf{I})$$



$$f(\mathbf{x}) := \sin(\mathbf{x} + \boldsymbol{\phi})^\top \mathbf{a}$$
$$\mathbf{a} \sim \mathcal{N}(\mathbf{1}, c_1\mathbf{I})$$
$$\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, c_2\mathbf{I})$$

Experimental Setup

- Estimator: model $\hat{M}$ trained with a given dataset $\mathcal{Z} := \{\mathbf{x}_i, y_i\}_{i=0}^{N}$
- Performance: expected error over the task distribution $p_{\mathcal{T}}$ and data distribution $p_{\mathcal{Z}}$

$$\mathcal{E} := \iiint (\hat{\mathcal{M}}(\mathbf{x}) - y)^2 p(\mathbf{x}, y) p_{\mathcal{Z}} p_{\mathcal{T}} d\mathbf{x} dy d\mathcal{Z} d\mathcal{T} \tag{1}$$

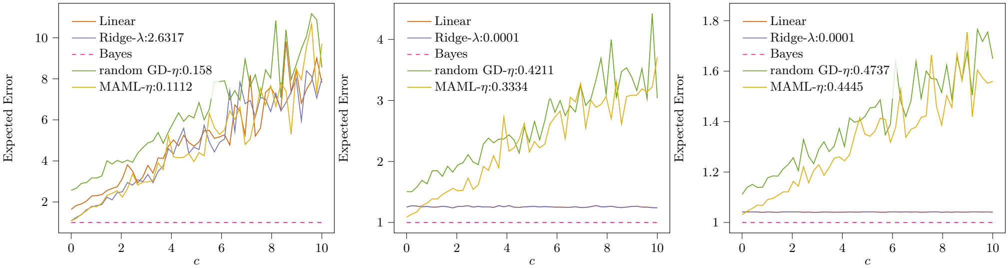
Experimental Setup

Models under investigation;

- Linear and Kernel Ridge Regression
- MAML (initialized from $\mathbf{w}_{\text{meta}}$) with limited adaptation
- Randomly initialized gradient descent

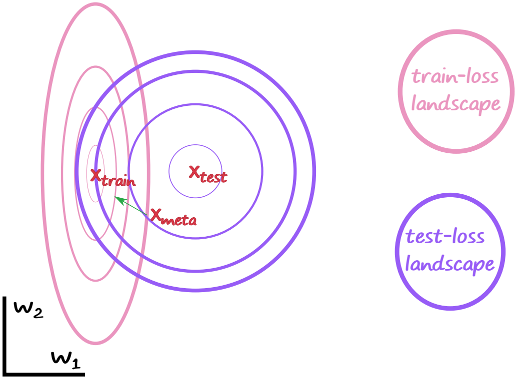
(Some) Results

- Task Variance (c) for 1D-Linear Problem with $\sigma = 1$, $m = 0$, $k = 1$, $c = 1$, $n_{iter} = 1$ and $N = 1, 10, 50$



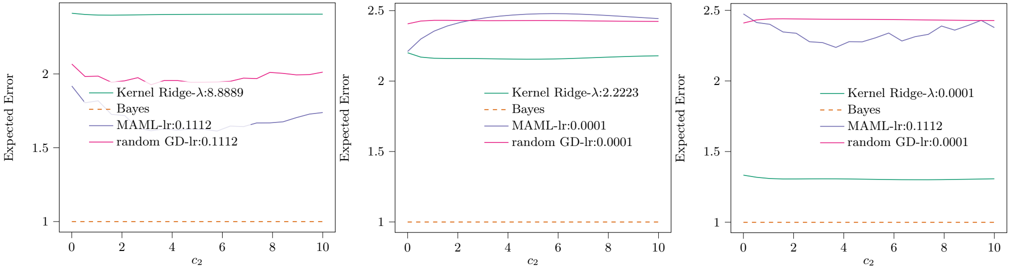
(Some) Results

- What is happening?



(Some) Results

- Task Variance (c_2) for 1D-Nonlinear Problem with $\sigma = 1$, $k = 1$, $c_1 = 2$, $n_{iter} = 10$ $N = 1, 10, 50$



Conclusions

- A single-task learner can outperform MAML with limited gradient step adaptation in expectation.
- Small task variance is crucial for MAML performance in expectation.
- A similar study for supervised benchmark datasets can be done to understand the generalization performance of MAML and its variants better.

Thanks for your attention!