BNAIC/BeNeLearn 2022

November 6, 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]

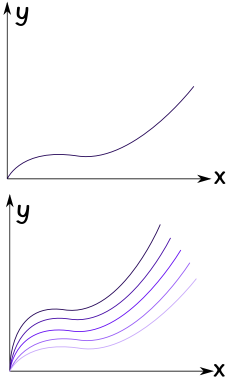
[1] Delft University of Technology, Pattern Recognition and Bioinformatics Labratory

Outline

- Introduction (Meta-Learning & MAML)
- Experimental Setup
- (Some) Results
- Conclusions

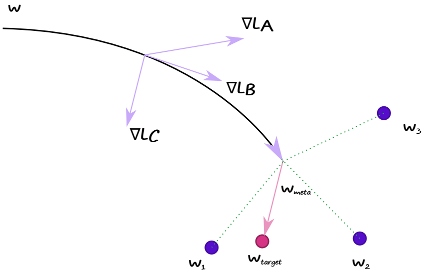
Meta-Learning

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



MAML[1]

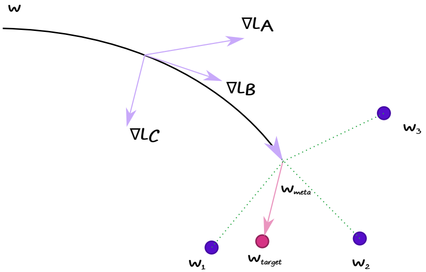
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps



[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]

MAML[1]

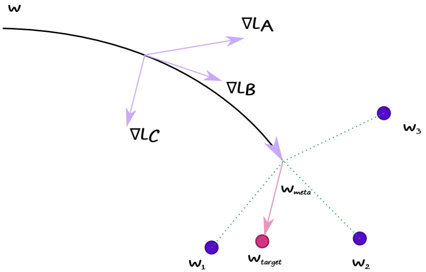
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps



[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]

MAML[1]

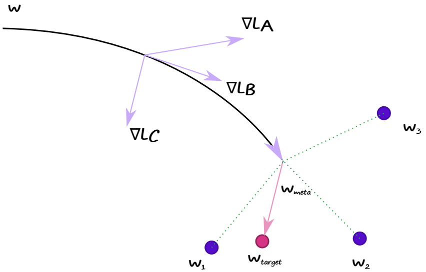
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps



[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]

MAML[1]

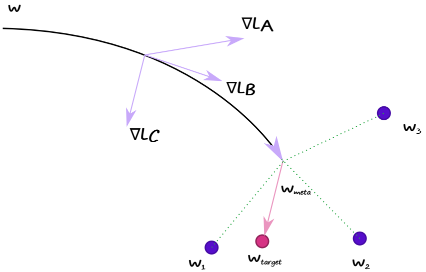
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps



[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]

MAML[1]

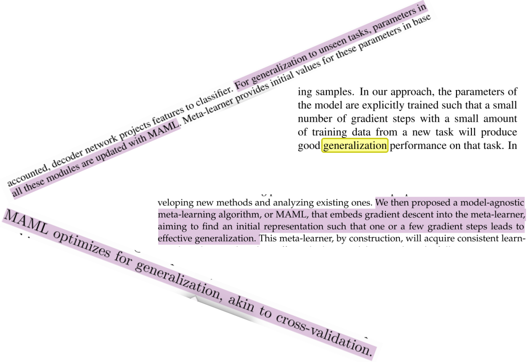
- Assume you have task distribution $p_{\mathcal{T}}$
- Sample a batch of tasks $\{\mathcal{T}_i\}_{i=1}^{M}$
- Provide an initialization for model parameters
- Get a target task $\mathcal{T}_{\text{target}}$
- Adaptation to a target task with limited gradient steps



[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]

MAML[1]

- Generalization
- Quick adaptation is not needed by many settings



[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]

MAML[1]

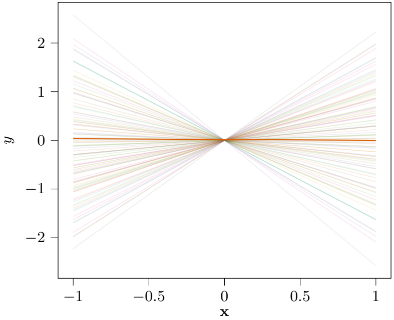
AIM: Investigate the effect of gradient step limitation on generalization performance!

[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]

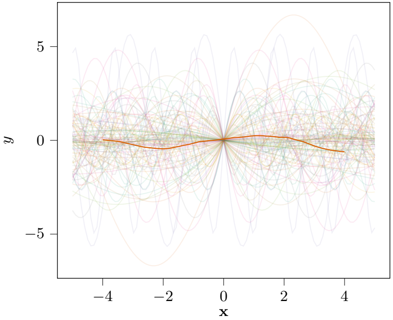
Experimental Setup

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

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

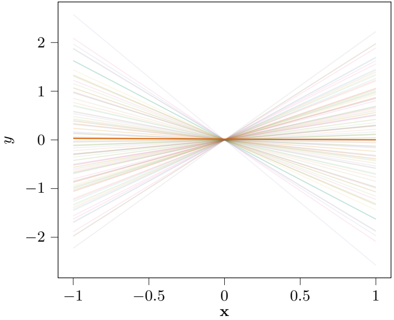


$f(\mathbf{x}) := \mathbf{x}^\mathsf{T}\mathbf{a}$

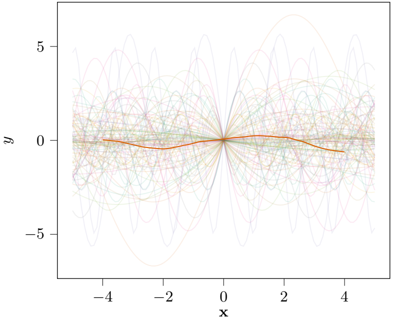


$f(\mathbf{x}) := \sin(\mathbf{x} + \boldsymbol{\phi})^\mathsf{T}\mathbf{a}$

Experimental Setup



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



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

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 $p_{\mathcal{Z}}$

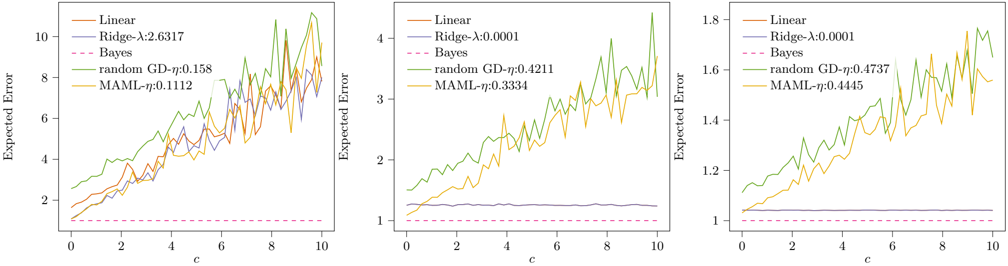
$$\mathcal{E} := \iiint (\hat{\mathcal{M}}(\mathbf{x}) - y)^2 p(\mathbf{x}, y) p_{\mathcal{Z}} p_{\mathcal{T}} d\mathbf{x} dy d\mathcal{Z} d\mathcal{T} \tag{1}$$

Models under investigation;

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

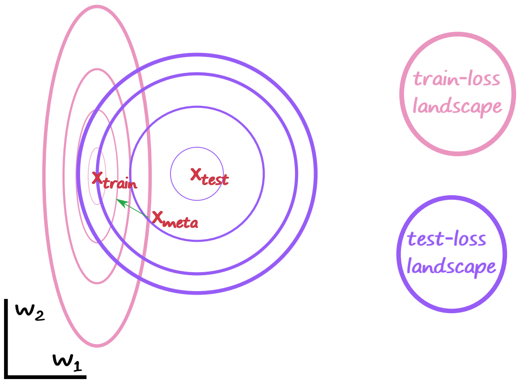
(Some) Results

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



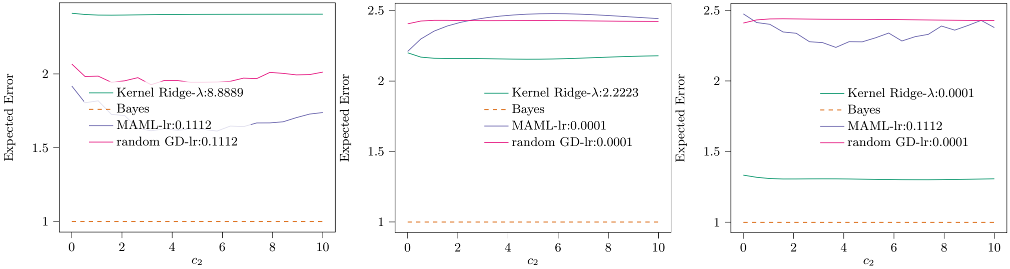
(Some) Results

- What is happening?



(Some) Results

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



Conclusions

After our detailed investigation

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

Thanks for your attention!