# MAML

Bingjie Yan

July 23, 2021

# MAML

Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks

MAML = Model-Agnostic Meta-Learning

也就是与模型无关的元学习，能够快速适应新的学习任务

Chelsea Finn, Pieter Abbeel, Sergey Levine

University of California, Berkeley

ICML 2017

# MAML 问题提出

- Few-shot Learning
- Meta-Learning
- Transfer Learning
- 区别与联系?

小样本的元学习问题的设立就是要找到一个 model，能够利用较少的数据，经过较少的迭代，能够快速适应到一个新任务中

# 几个元学习中的概念

- Support Set(Training Set, 小朋友手中的卡片)
- Query Set(Test Set, Target Set, 小朋友看到的事物)
- N-way K-shot(N classes, K samples) -> 一个 task

# MAML 核心内容

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{1}$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{2}$$

优化目标:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{3}$$

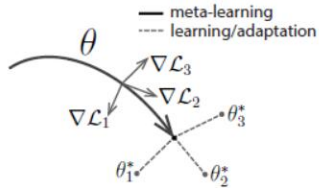



*Figure 1.* Diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation $\theta$ that can quickly adapt to new tasks.

# How?

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{4}$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{5}$$

第一个式子代入到第二个式子中，因为有两个 $\nabla$，也就是两层的 gradient，所以就相当于复合函数求导，$f(g(x)) = f'(g(x))g'(x)$，qrt=query set，spt=support set，最后得到

$$\theta \leftarrow \theta - \beta [\nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}^{qrt}_{\mathcal{T}_i}(\theta - \alpha \nabla_\theta \mathcal{L}^{spt}_{\mathcal{T}_i}(\theta)) \cdot (I - \alpha \nabla^2_\theta \mathcal{L}^{spt}_{\mathcal{T}_i}(\theta))] \tag{6}$$

# How?

前页中出现的二阶 gradient 是直接用一阶近似 $(f(x) = g(x_0) + g'(x_0)(x - x_0))$ 直接给忽略掉了二阶导数

更深一层的理解是，如果将 $\theta$ 展开，对每个元素求解可以得到类似的结果，假设 $\theta$ 各元素为 $\theta^j$，$\theta_i'$ 的各元素为 $\theta_i'^k$，后面二阶导数式子产生的其实就是一个偏导的方阵 $\frac{\partial \theta_i'}{\partial \theta}$ 又有 $\theta_i$ 和 $\theta$ 的关系 $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ 很容易得到

$$\frac{\partial \theta_i'^k}{\partial \theta^j} = \begin{cases} 1 - \alpha \frac{\partial^2 \mathcal{L}_{\mathcal{T}_i}}{\partial \theta^j \theta^k} \approx 1 & j = k \\ -\alpha \frac{\partial^2 \mathcal{L}_{\mathcal{T}_i}}{\partial \theta^j \theta^k} \approx 0 & j \neq k \end{cases} \tag{7}$$

在忽略二阶导数以后，外层的的偏导数 $\frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta^j}$ 就变成了

$$\frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta^j} = \sum_k \frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_i'^k} \frac{\partial \theta_i'^k}{\partial \theta^j} \approx \frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_i'^j} \tag{8}$$

其实也就是在外层优化的时候直接对 $\theta_i'$ 求导即可，无需对起始的 $\theta$ 进行，这样做可以大大降低计算的复杂度

# MAML 算法流程

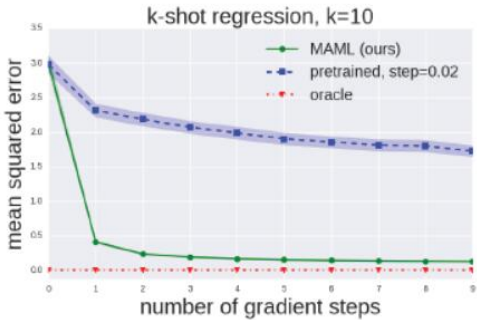
**Algorithm 1** Model-Agnostic Meta-Learning

**Require:** $p(\mathcal{T})$: distribution over tasks
**Require:** $\alpha$, $\beta$: step size hyperparameters
1: randomly initialize $\theta$
2: **while** not done **do**
3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
4: **for all** $\mathcal{T}_i$ **do**
5: Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to $K$ examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
7: **end for**
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
9: **end while**

# 实验

实验部分主要在验证三个问题：

- MAML 能否实现新任务的快速学习？
- MAML 是否可以用于元学习在多个不同的领域，包括监督回归、分类和强化学习？
- MAML 学习的模型能否在后续的梯度更新继续提升？



# 实验

| Omniglot (Lake et al., 2011) | 5-way Accuracy | | 20-way Accuracy | |
|---|---|---|---|---|
| | 1-shot | 5-shot | 1-shot | 5-shot |
| MANN, no conv (Santoro et al., 2016) | 82.8% | 94.9% | – | – |
| **MAML, no conv (ours)** | **89.7 ± 1.1%** | **97.5 ± 0.6%** | – | – |
| Siamese nets (Koch, 2015) | 97.3% | 98.4% | 88.2% | 97.0% |
| matching nets (Vinyals et al., 2016) | 98.1% | 98.9% | 93.8% | 98.5% |
| neural statistician (Edwards & Storkey, 2017) | 98.1% | 99.5% | 93.2% | 98.1% |
| memory mod. (Kaiser et al., 2017) | 98.4% | 99.6% | 95.0% | 98.6% |
| **MAML (ours)** | **98.7 ± 0.4%** | **99.9 ± 0.1%** | **95.8 ± 0.3%** | **98.9 ± 0.2%** |

| MiniImagenet (Ravi & Larochelle, 2017) | 5-way Accuracy | |
|---|---|---|
| | 1-shot | 5-shot |
| fine-tuning baseline | 28.86 ± 0.54% | 49.79 ± 0.79% |
| nearest neighbor baseline | 41.08 ± 0.70% | 51.04 ± 0.65% |
| matching nets (Vinyals et al., 2016) | 43.56 ± 0.84% | 55.31 ± 0.73% |
| meta-learner LSTM (Ravi & Larochelle, 2017) | 43.44 ± 0.77% | 60.60 ± 0.71% |
| **MAML, first order approx. (ours)** | **48.07 ± 1.75%** | **63.15 ± 0.91%** |
| **MAML (ours)** | **48.70 ± 1.84%** | **63.11 ± 0.92%** |

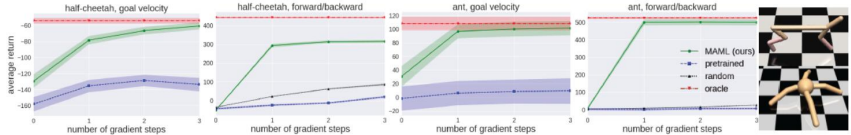



*Figure 5.* Reinforcement learning results for the half-cheetah and ant locomotion tasks, with the tasks shown on the far right. Each gradient step requires additional samples from the environment, unlike the supervised learning tasks. The results show that MAML can adapt to new goal velocities and directions substantially faster than conventional pretraining or random initialization, achieving good performs in just two or three gradient steps. We exclude the goal velocity, random baseline curves, since the returns are much worse ($< -200$ for cheetah and $< -25$ for ant).

# Datasets

Omniglot(likes MNIST):
20 instances of 1623 characters from 50 different alphabets
Mini-imagenet:
Total: 100 classes, 600 samples, 84 x 84 image size
64 training classes, 12 validation classes, and 24 test classes

# Reproduce

Result on Mini-imagenet

| | 5-way 1-shot | 5-way 5-shot |
|---|---|---|
| Meta-LSTM | 43.44% | 60.60% |
| MAML | 48.07±1.75% | 63.15±0.91% |
| mine | 43.82% | 59.40% |

# MAML v.s. Transfer Learning

MAML:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{9}$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \tag{10}$$

Transfer Learning(Pre-training):

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} l(\theta) \tag{11}$$

这里可以看出，Meta-Learning 更加关注这个模型的潜力，也就是模型后续的走向；而 Transfer Learning 更加关注与模型当前的表现

*Thanks!*