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

Tanguy Kerdoncuff

February 2019

1 Briefly summarize the main idea of the proposed method - MAML.

MAML is an algorithm that has been designed to solve the few shot learning problem in a learning to learn context. It can be plug with every gradient descent base algorithm (a Neural Network for example). MAML doesn't learn to solve each task but learn parameters which can be easily adapted to solve a new task with one (or few) gradient descent step.

2 Where the computational complexity of MAML update rule comes from? Write down the version of equation (1) modified to reduce its computational complexity.

The computational complexity of MAML comes from the double derivative (Hessian matrix of size $|\theta| \times |\theta|$) that needs to be computed to apply the gradient descent in the meta-objective. Let make a first order approximation on the first derivative. Let θ_j be a parameter. Let for all $k \in [\![0, |\theta|]\!]$ e_k be the vector of size $|\theta|$ full of 0 except with a 1 at the position k. Let h > 0 be a small value.

$$\frac{\partial \sum_{\mathcal{L}_{\mathcal{T}_i \sim p(\mathcal{T})}} \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_j}(\theta_i') = \sum_{\mathcal{L}_{\mathcal{T}_i \sim p(\mathcal{T})}} \frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_j}\left(\theta - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(\theta + he_k) - \mathcal{L}_{\mathcal{T}_i}(\theta)}{h}\right)_{k \in [\![0,|\theta|]\!]}\right)$$

Let apply the chain rule to see clearly that we have avoided the computation of the second derivative

$$= \sum_{\mathcal{L}_{\mathcal{T}_i \sim p(\mathcal{T})}} \sum_{k=1}^{|\theta|} \frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_k - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(\theta + he_k) - \mathcal{L}_{\mathcal{T}_i}(\theta)}{h}\right)}\left(\theta - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(\theta + he_l) - \mathcal{L}_{\mathcal{T}_i}(\theta)}{h}\right)_{l \in [\![0,|\theta|]\!]}\right)$$

$$\times \frac{\partial ._k - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(. + he_k) - \mathcal{L}_{\mathcal{T}_i}(.)}{h}\right)}{\partial \theta_j}(\theta)$$

$$= \sum_{\mathcal{L}_{\mathcal{T}_i \sim p(\mathcal{T})}} \sum_{k=1}^{|\theta|} \frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_k - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(\theta + he_k) - \mathcal{L}_{\mathcal{T}_i}(\theta)}{h}\right)}\left(\theta - \alpha\left(\frac{\mathcal{L}_{\mathcal{T}_i}(\theta + he_l) - \mathcal{L}_{\mathcal{T}_i}(\theta)}{h}\right)_{l \in [\![0,|\theta|]\!]}\right)$$

$$\left(\frac{\partial \theta_k}{\partial \theta_j} - \frac{\alpha}{h}\frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_j}(\theta + he_k) + \frac{\alpha}{h}\frac{\partial \mathcal{L}_{\mathcal{T}_i}}{\partial \theta_j}(\theta)\right)$$

The only thing that need to be computed here is the derivative of $\mathcal{L}_{\mathcal{T}_i}$ according to all his components and then need to be evaluated at different points, there is no more double derivative. The speedup in the computation is not completely clear here and would highly depend on the implementation.

3 What approach is used by authors in order to reduce overfitting of the resulting models? Where it occurs in Algorithms 2 and 3?

To reduce the overfitting the author uses two different sample for each update. It is clear in the Algorithm 2 (or 3). For each task $\mathcal{T}_i$ with i an integer the computation of the gradient descent is associated with K data-points (or trajectories) line 5. However in the line 8 there is another sample to compute the meta-update. This will ensure some generalization. The algorithm is train to generalize during the training part.

4 Propose an idea of solving the following problem using MAML: Given a dataset composed of images of Dogs and Cats, train a binary classificator appropriate for fast adaptation to tasks of the form "Dogs vs. SomeoneElse". Is your solution theoretically justified (in accordance with the paper)? If it is not, why? If applying MAML for this problem is not possible, explain why it is so.

At first sign this situation seems not great for the MAML algorithm, it is not a classical meta-learning problem because of the datasets available. If the goal is to solve the "Dog vs someone else" problem, we expect to have more than one dataset ("Dog vs Cat"). The MAML algorithm does not take advantage of large cat dataset but more of different datasets.

We can still apply MAML algorithm but instead of drawing a random task we always draw a "Dog vs Cat" task which will give you a completely biased algorithm. Maybe this algorithm will work with only one task which is better than nothing. If it is allowed, a good idea to improve the program will be to define new classes for the cat dataset like the race of cat. This would still be a very biased selection of task. However this can improve MAML because this will force the algorithm to be ready to more different tasks.