Review of "Model Agnostic Meta Learning for Fast Adaptation of Deep Networks"

Gaurav Iyer

August 2019

1 Problem Statement and Aim

Humans can easily recognize an object, and distinguish them from other objects with exposure to very few examples, and quite quickly. In the current era of artificial intelligence, there is a focus on trying to make our models learn the way we learn, and meta learning is the way to teach our models to learn with relatively fewer examples and to generalise to different tasks faster. The paper "Model Agnostic Meta Learning for Fast Adaptation of Deep Networks" proposes a meta learning algorithm that is "model agnostic" i.e. the model in use is irrelevant to the algorithm, as long as it is trained using gradient descent (which is quite often the case). Also, it is applicable to a variety of learning problems, such as regression, supervised classification, and even reinforcement learning problems.

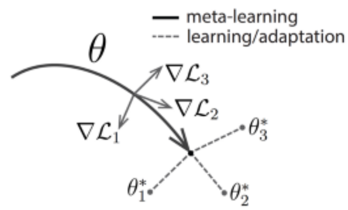
The proposed method aims to optimize the model parameters such that one or a small number of gradient steps on a new task will produce large changes in the weights, bringing it closet to properly learning the task, but without over-fitting it.

2 Assumptions

No assumptions are made on the form of the model, other than to assume that is parametrised by some parameter vector theta. It is also assumed that the loss function is smooth enough to make use of gradient based learning techniques.

3 Basic Overview of the Algorithm (MAML)

In essence, MAML provides an initialization of a model's parameters to optimally learn a new task faster, and with fewer number of data points and gradient steps. At the same time, it aims to prevent the overfitting that occurs when making use of small datasets.



When using meta-learning, we make use of a learner and a meta-learner. The meta learner trains the learner (or the model) on multiple different tasks in order to achieve this optimal initialisation. Then, depending on the task, the learner is fine tuned with a small number of examples to learn that specific task. In the above diagram, we can observe that the model parameter vector theta is equally close to all three parameter vectors theta1, theta2, and theta3, which is what the algorithm aims to achieve. As a result, even a small change in the parameters of theta will optimize the loss function for that specific task.

4 Experimental Evaluation and Results

4.1 Regression

Task: Regressing from the input to the output of a sine wave, where the amplitude and phase are varied.

Loss Function: Mean Squared Error

To evaluate performance, a single meta-learned model is fine tuned on a varying number of examples, and then compared to certain set baselines. Additionally, the number of data points given to the model for fine tuning are also varied.

Results: MAML is able to estimate parts of the sine curve where data points are missing, showing that it has learned the structure. It is able to improve with additional gradient steps, and at the same time overfit to the small sized dataset. MAML is able to optimize parameters in such a way that fast adaptation is possible, and is sensitive to loss functions.

4.2 Classification

Task: Classification on MiniImage and Omniglot datasets. These datasets are commonly used to evaluate few shot learning algorithms and benchmarks.

Loss Function: Cross Entropy

Experimental evaluation involves fast learning of N way classification. The problem of N way classification is set up as follows: select N unseen classes, provide the model with K different instances of each of the N classes, and evaluate the model's ability to classify new instances within the N classes.

Results:

Omniglot (Lake et al., 2011)	5-way Accuracy 1-shot	5-shot	20-way Accuracy 1-shot	5-shot
MANN, no conv (Santoro et al., 2016)	82.8%	94.9%	–	–
MAML, no conv (ours)	$\mathbf{89.7 \pm 1.1\%}$	$\mathbf{97.5 \pm 0.6\%}$	–	–
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)	$\mathbf{98.7 \pm 0.4\%}$	$\mathbf{99.9 \pm 0.1\%}$	$\mathbf{95.8 \pm 0.3\%}$	$\mathbf{98.9 \pm 0.2\%}$

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

Reinforcement learning experiments show that MAML can adapt to new goals substantially faster than conventional pretraining or random initialisations.