ONLINE META-LEARNING

Chelsea Finn Aravind Rajeswaran Sham Kakade Sergey Levine

SAI SARAN ANAMANAMUDI

University at Buffalo The State University of New York

CONTENTS

Introduction

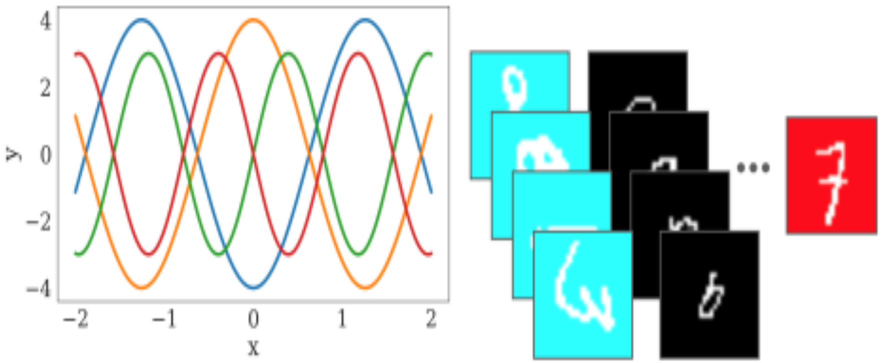
Online meta learning is the process of a machine learning algorithm continuously adjusting to new data and using that knowledge to update its parameters and produce more accurate predictions. which means that the algorithm picks up new information and adjusts to it in real-time. In online meta learning, the algorithm continuously picks up new information from fresh data and applies that knowledge to adjust its parameters, allowing it to gradually improve its predictions.

For example, In reinforcement learning the agent interacts with the environment and gathers input in the form of rewards and punishments. It then uses this data to update its model and enhance its functionality.

Example

Sinusoid functions of naive training on prior tasks fails, Figure also shows colored MNIST digits with different backgrounds. Suppose we've seen MNIST digits with various colored backgrounds, and then observe a "7" on a new color. We might conclude from training on all of the data seen so far that all digits with that color must all be "7."

In fact, this is an optimal conclusion from a purely statistical standpoint. However, if we understand that the data is divided into different tasks and take note of the fact that each task has a different color, a better conclusion is that the color is irrelevant. Training on all of the data together, or only on the new data, does not achieve this goal.



Foundations

- Few-Shot Learning
- MAML
- Online Learning

Few-Shot Learning

In the few-shot supervised learning setting, we are interested in a family of tasks, where each task T is associated with a notional and infinite-size population of input-output pairs. In the few-shot learning, the goal is to learn a task while accessing only a small, finite-size labeled dataset $D_i := \{x_i , y_i\}$ corresponding to task Ti. If we have a predictive model, h(·; w), with parameters w, the population risk of the model is

$$f_i(\mathbf{w}) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{T}_i}[\ell(\mathbf{x},\mathbf{y},\mathbf{w})],$$

Few-Shot Learning

where the expectation is defined over the task population and L is a loss function, such as the square loss or cross entropy between the model prediction and the correct label. An example of L corresponding to squared error loss is

$$\ell(\mathbf{x}, \mathbf{y}, \mathbf{w}) = ||\mathbf{y} - \boldsymbol{h}(\mathbf{x}; \mathbf{w})||^2.$$

Meta-Learning and MAML

- Meta-learning or learning to learn aims to effectively bootstrap from a set of tasks to learn faster on a new task. It is assumed that tasks are drawn from a fixed distribution,
- Meta-learning algorithms attempt to find a model using the M training tasks, such that when D_j is revealed from the test task, the model can be quickly updated to minimize $f_j(w)$.

Meta-Learning and MAML

- Model-agnostic meta-learning (MAML) does this by learning an initial set of parameters w(MAML), such that at meta-test time, performing a few steps of gradient descent from w(MAML) using D_j minimizes $f_j(\cdot)$. To get such an initialization, at meta-training time, MAML solves the optimization problem.

$$\mathbf{w}_{\mathrm{MAML}} := \arg\min_{\mathbf{w}} \frac{1}{M} \sum_{i=1}^{M} f_i\big(\mathbf{w} - \alpha \nabla \hat{f}_i(\mathbf{w})\big).$$

Online Learning

- In the online learning setting, an agent faces a sequence of loss functions $\{f_t\}^{\infty}_{t=1}$, one in each round t. These functions need not be drawn from a fixed distribution and could even be chosen adversarial over time. The goal for the learner is to sequentially decide on model parameters $\{w_t\}^{\infty}_{t=1}$ that perform well on the loss sequence. In particular, the standard objective is to minimize some notion of regret defined as the difference between our learner's loss. The most standard notion of regret is to compare to the cumulative loss of the best fixed model in hindsight:

$$\mathrm{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}} \sum_{t=1}^{T} f_t(\mathbf{w}).$$

Online Learning

The goal in online learning is to design algorithms such that this regret grows with T as slowly as possible. In particular, an agent (algorithm) whose regret grows sub-linearly in T is non-trivially learning and adapting. One of the simplest algorithms in this setting is follow the leader (FTL) which updates the parameters as:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \sum_{k=1}^{t} f_k(\mathbf{w}).$$

The Online Meta-Learning Problem

The goal for the agent is to minimize regret over the rounds. A highly ambitious comparator is the best meta-learned model in hindsight. Let $\{w_t\}_{t=1}^{T}$ be the sequence of models generated by the algorithm. Then, the regret we consider is:

$$\text{Regret}_T = \sum_{t=1}^{T} f_t\big(\boldsymbol{U}_t(\mathbf{w}_t)\big) - \min_{\mathbf{w}} \sum_{t=1}^{T} f_t\big(\boldsymbol{U}_t(\mathbf{w})\big).$$

Algorithm and Analysis

The aim of our experimental evaluation is to study the following questions:

1) can online meta-learning (and specifically FTML) be successfully applied to multiple non-stationary learning problems

2) does online meta-learning (FTML) provide empirical benefits over prior methods

By compare to the following algorithms: (a) Train on everything (TOE) trains on all available data so far (including D_t at round t) and trains a single predictive model. This model is directly tested without any specific adaptation since it has already been trained on D_t. (b) Train from scratch, which initializes w_t randomly, and finetunes it using D_t. (c) Joint training with fine-tuning, which at round t, trains on all the data jointly till round t − 1, and then finetunes it specifically to round t using only Dt. This corresponds to the standard online learning approach where FTL is used, followed by task-specific fine-tuning.

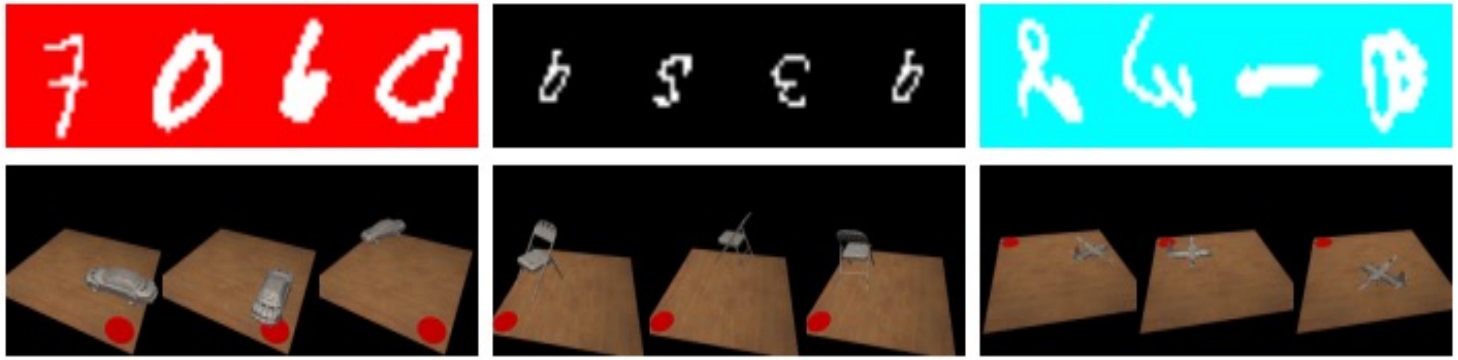
Algorithm

Algorithm 1 Online Meta-Learning with FTML

Input: Performance threshold of proficiency, γ
randomly initialize $\mathbf{w}_1$
initialize the task buffer as empty, $\mathcal{B} \leftarrow [\,]$
for $t = 1, \ldots$ **do**
  initialize $\mathcal{D}_t = \emptyset$
  Add $\mathcal{B} \leftarrow \mathcal{B} + [\,\mathcal{T}_t\,]$
  while $|\mathcal{D}_{\mathcal{T}_t}| < N$ **do**
    Append batch of n new datapoints $\{(\mathbf{x}, \mathbf{y})\}$ to $\mathcal{D}_t$
    $\mathbf{w}_t \leftarrow \texttt{Meta-Update}(\mathbf{w}_t, \mathcal{B}, t)$
    $\tilde{\mathbf{w}}_t \leftarrow \texttt{Update-Procedure}(\mathbf{w}_t, \mathcal{D}_t)$
    if $\mathcal{L}(\mathcal{D}_t^{\text{test}}, \tilde{\mathbf{w}}_t) < \gamma$ **then**
      Record efficiency for task $\mathcal{T}_t$ as $|\mathcal{D}_t|$ datapoints
    end if
  end while
  Record final performance of $\tilde{\mathbf{w}}_t$ on test set $\mathcal{D}_t^{\text{test}}$ for task t.
  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
end for

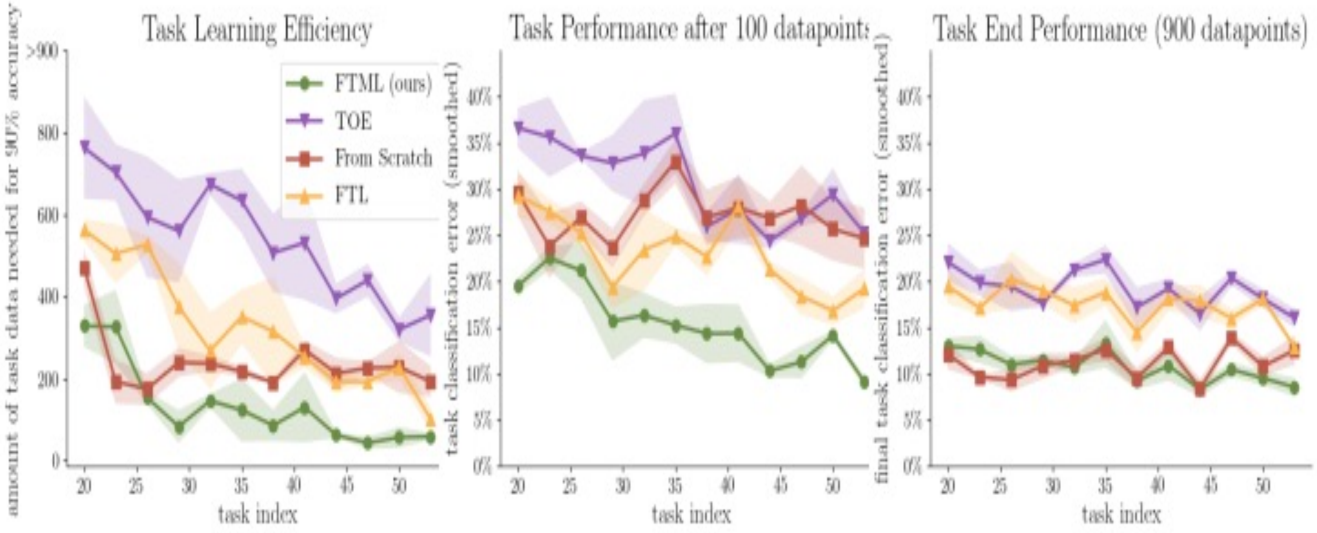
Five-Way CIFAR-100

Illustration of three tasks for Rainbow MNIST (top) and pose prediction (bottom). Rainbow MNIST includes different rotations, scaling factors, and background colors. For the pose prediction tasks, the goal is to predict the global position and orientation of the object on the table. Cross-task variation includes varying 50 different object models within 9 object classes, varying object scales, and different camera viewpoints

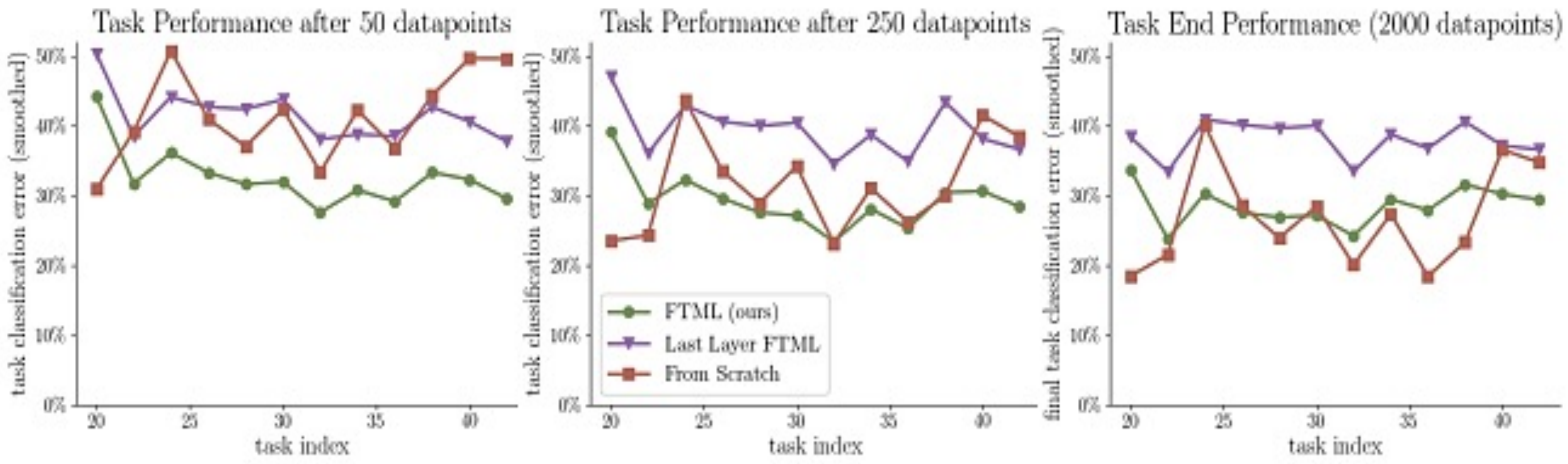


Rainbow MNIST Results

Left: amount of data needed to learn each new task. Center: task performance after 100 datapoints on the current task. Right: The task performance after all 900 datapoints for the current task have been received. Lower is better for all plots, while shaded regions show standard error computed using three random seeds. FTML can learn new tasks more and more efficiently as each new task is received, demonstrating effective forward transfer

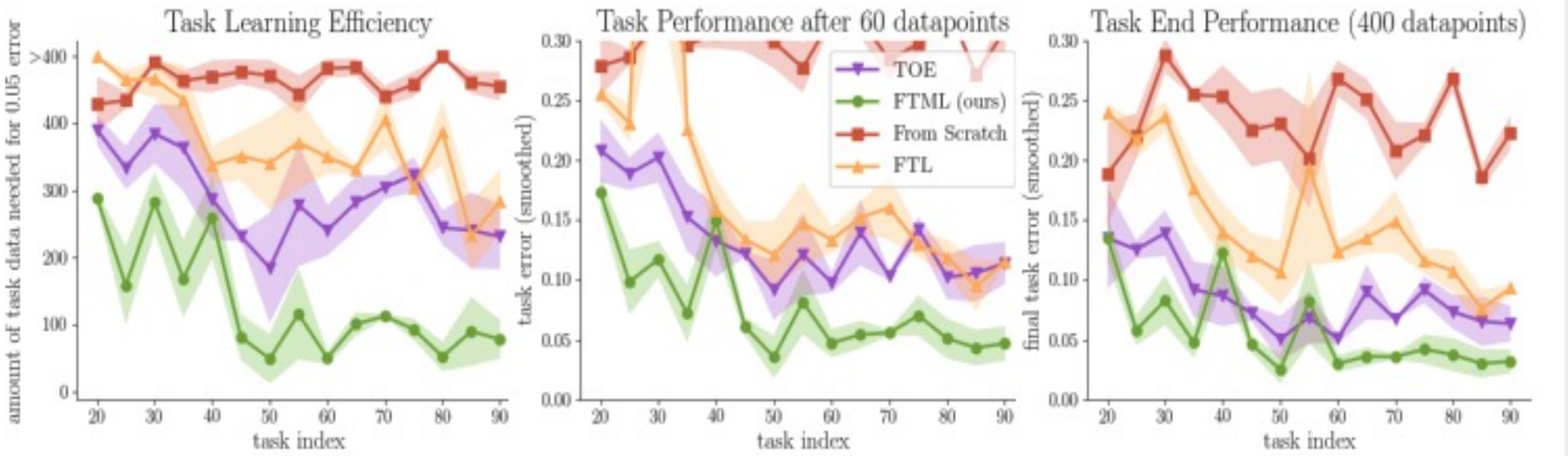


Online CIFAR-100 Results



evaluating task performance after 50, 250, and 2000 datapoints have been received for a given task. We see that FTML learns each task much more efficiently than models trained from scratch, while both achieve similar asymptotic performance after 2000 datapoints. We also observe that FTML benefits from adapting all layers rather than learning a shared feature space across tasks while adapting only the last layer.

Object Pose Prediction Results



Left: we observe that online meta-learning generally leads to faster learning as more and more tasks are introduced, learning with only 10 datapoints for many of the tasks. Center & right, we see that meta-learning enables transfer not just for faster learning but also for more effective performance when 60 and 400 datapoints of each task are available. The order of tasks is randomized, leading to spikes when more difficult tasks are introduced. Shaded regions show standard error across three random seeds

Conclusion

- We concentrated our analysis on the case where the update procedure Ut, inspired by MAML, corresponds to one step of gradient descent. However, in practice, many works with MAML (including our experimental evaluation) use multiple gradient steps in the update procedure, and backpropagate through the entire path taken by these multiple gradient steps. Analyzing this case, and potentially higher order update rules will also make for exciting future work
- Primarily aimed to discern if it is possible to meta-learn in a sequential setting. For this purpose, proposed the FTML template algorithm which draws inspiration from FTL in online learning. It is well known that FTL has poor computational properties, since the computational cost of FTL grows over time as new loss functions are accumulated. Further, in many practical online learning problems, it is challenging (and sometimes impossible) to store all datapoints from previous tasks. The method can effectively learn nearly 100 Online Meta-Learning tasks in sequence without significant burdens on compute or memory, scalability remains a concern.

Thank You